Tampere University

Haifeng Xu

AN EVALUATION OF ONE CLASS CLASSIFIER ON GENE EXPRESSION DATA

Information Technology and Communication Science
Master's Thesis
August 2019

ABSTRACT

Haifeng Xu: An Evaluation of One Class Classifier on Gene Expression Data
Master's Thesis
Tampere University
Master of Science (Technology)
August 2019

It is not rare that medical data has imbalanced classes. This problem causes many difficulties when diagnosing rare diseases or cancer subtypes by machine learning tools, since traditional binary or multi-class classifiers lack the ability to classify imbalanced data. Therefore, One-Class Classifiers (OCC), the machine learning methods that only use data from one class, becomes one possible option. Our study evaluates ν-SVM, one of the most commonly used One-Class methods, on four microarray datasets of Breast Cancer and Diffuse large B-cell lymphoma (DLBCL). Each cancer is labelled into different subtypes. We compared OCC with binary SVM and studied how the imbalance between the classes affects the results. The results show that ν-SVM performs better than binary SVM when the data classes are extremely imbalanced on these datasets.

Keywords: Machine Learning, One-Class SVM, Bioinformatics, Microarray, Cancer Classification

PREFACE

This thesis is written for the Master's Degree in Science (Technology) of Tampere University. The thesis is about my personal research project supported by Medical Bioinformatics Centre, Turku Bioscience Centre, University of Turku and Åbo Akademi University.

Acknowledgements to Professor Tapio Elomaa, PhD, Tampere University, who has supervised this thesis. Thanks for his great advice about thesis writing. Acknowledgements to Adjunct Professor Laura Elo, PhD, University of Turku, the group leader of Medical Bioinformatics Centre, who has offered many guidances during the study. Acknowledgements to Post-doc Mikko Venäläinen, PhD, who guided me in the entire study with his knowledge and patience. Thanks to all the professors and teachers who taught me courses in Tampere University in the past three years.

Tampere, 8th August 2019

Haifeng Xu

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PROGRAMS AND ALGORITHMS

1 INTRODUCTION

Nowadays, pattern recognizing of medical data becomes very important because of the vigorous increasing of data size. Among them, machine learning, a group of study about algorithms and statistical models, has shown more and more influence. Generally speaking, machine learning is a method (or a group of methods), that allows one to build classification or regression models only by a pre-selected algorithm and the data itself. Since it does not require any explicit instructions from users, machine learning can usually find the hidden patterns that human cannot from the data.

Though it works well in many situations, there is still a serious problem when applying machine learning methods on medical data. In medical situations, it is very common that the two classes (patients vs healthy controls, or one subtype vs another) are not in balance. On one hand, practitioners cannot collect positive samples of rare diseases since they are not able to find patients. For example, a cancer may have 95% common subtype and 5% rare subtype, whose patients need special treatment. Due to the restriction of biological experiment (expensive and difficult to find volunteers), it is already difficult to collect data from the common subtype. As for the rare subtype, there could be only a few samples, or even less. On the other hand, for those common diseases with high diagnostic costs, it is not easy to collect negative samples either. For binary or multi-class machine learning methods, this is a challenge. In his review study, Khan shows that the performances of traditional multi-class classifiers suffer from the high imbalance between the classes [36].

Another problem with binary or multi-class classification methods is that they usually classify a new data point into one of the pre-defined categories [36]. But if a classifier tries to classify new samples from an entirely different domain, the results will always be wrong. For instance, imagine a multi-class model that was trained by cats, cows and pandas. What would happen if it tries to classify a car? It would return an animal category anyway, which is clearly wrong. In practice, a classification task is usually not to allocate new data points into several categories. Instead, the classifier should decide if this new data point belongs to a specific class or not [36].

All the problems above also frequently exist in medical studies. For instance, a cancer may have several subtypes but only one of them can achieve a high 5-year survival rate [69]. Additionally, there could be undiscovered subtypes. Therefore, if one wants to build a classifier to diagnose the "safe" subtype, one may need an approach that can learn the decision function only by the data of a single class. We expect this one-class approach can learn the decision boundary from the safe subtype, then can still determine if a new patient has any other "unsafe" subtype, even this subtype is unknown yet.

For above reasons, several **One-Class Classification (OCC)** methods have been developed in recent years. Comparing to binary approaches, OCC only requires data from the "interesting class". This allows for munificent savings in memory space and computation time, while keeping a comparable level of performance [61]. Among all OCC methods, Support Vector Machine based approaches (OSVM) have generated the most amount of applications. Because of their advancements, applicability, and the huge amount of applications, OSVM methods show their abilities that can be a separate research area in their own right [36]. The first OSVM approach, Support Vector Data Description (SVDD), was invented by Tax and Duin in 1999 [67]. Then in 2000, Schölkopf et al. published a new method, ν-SVM [59]. These two algorithms are compared and optimized in the past 20 years for several times. But in practice, they are usually used in text or documents classification. Only a few studies have applied OCC on medical data, such as applying on electroencephalography data, or MicroRNA of plants [23] [76]. Furthermore, only few studies applied OCC on gene expression data, although gene expression has been proved that it has a huge impact on many diseases. For example, certain genes are differentially expressed between multiple sclerosis patients and healthy people [35]. Therefore, it is potential to evaluate

how OCC works on gene expression data, in order to explore its clinical use.

In 2016, Sokolov et al. applied OCC on cancer subtype classification, which showed the potential use of OCC on gene expression data [63]. But some aspects in their study need more discussions. For instance, they evaluated the performance of OCC by *Area under the Curve (AUC)*. However, AUC cannot distinguish which class contributes more to the overall performance [22]. They did not discuss the effect of imbalance either, although some studies have pointed out that the skewness between classes can influence the classification results [36] [22]. Therefore, the purpose of this study is to evaluate how *one class classifier* works on *gene expression data*, especially when the data classes are imbalanced. Furthermore, a proper evaluation standard is needed. We also try to explore if OCC is applicable on cancer subtype diagnosing.

For above purposes, in this study, both ν-SVM and binary SVM are evaluated on four microarray datasets of two cancers, breast cancer and lymphoma. Each cancer has two labels, which are two different subtypes. For each cancer, we use one data set to train the classification models, and the other is for testing the performance. The training and test sets are uploaded on NCBI database by different organizations, which means these data sets are independent. Since they are independent, the performance would not be affected by experimental factors, such as the environment of laboratory or the operating habits of experimenters. We also have additional imbalanced tests to observe the influence of imbalance. In the imbalanced tests, the negative samples of the training set are randomly removed by 25%, 50%, 75%, and 90% respectively. Then we use the rest training samples to train the binary models, and compare their results with the performance of OCC.

In this study, we find that ν-SVM classifier has a comparable performance with binary SVM at the same level. When we remove negative samples from the training set, the performances of binary models reduce. The more negative samples we remove, the worse the binary models perform. Finally, the results of binary SVM become lower than OCC. Since OCC does not need negative data at all, it can be a better choice when the training set has only a few negative samples. We also discuss the possible threshold of the class ratio (Positive:Negatives) when OCC exceeds the binary models.

Chapter 2 of this thesis reviews the basic concept of OCC and other related studies. It also explains the concept of machine learning, basic mathematical expressions of the algorithms we use, and the related biology knowledge.

Chapter 3 introduces the methods and materials we use in this study. It includes the algorithm applications in R environment, the details of training and test data, the data pre-processing and feature selection, and the overall workflow.

Chapter 4 explains the results we have in this study, including the feature selection, kernel comparison and the imbalance test. It also discusses the limitation of this study, and the appropriate circumstances to use OCC on gene expression data.

Chapter 5 gives the overall conclusion, the possible improvement, and the challenges of this study. A quick glance of this study is also give here.

Other related materials are attached in the Chapter "Appendix".

2 THEORETICAL BACKGROUND

2.1 Related Work

The concept of OCC was mentioned for the first time in 1975 by Minter et al [46]. They proposed a classifier that only used data from "the class of interest". In the past 40 years, following terms were also used to describe OCC problems, such as *Single Class Classification* [49], *Outlier Detection* [56], *Concept Learning* [29], and *Novelty Detection* [7]. In 2006, Juszczak defined One-Class Classifiers as *"class descriptors that are able to learn restricted domains in a multi-dimensional pattern space using primarily just a positive set of examples"* [55].

In his review study, Khan divides OCC methods into two categories [36]:

1 *OSVM:* Support Vector Machine (SVM) based one-class methods

2 *Non-OSVM:* including many one-class machine learning techniques based on random forest, neural network, logistic regression, and others.

Although it seems unfair to regard SVM-based methods as an independent category, they do have shown that their advancements, applications, significance and difference, which made OSVM a separate research area [36].

As we mentioned in Chapter 1, the two main OSVM methods are SVDD and ν-SVM. The difference between them is how they separate the classes. SVDD constructs a hyper-sphere around the data, while ν-SVM creates a hyper-plane to separate the origin and the region that contains data [36]. These two algorithms are optimized for many times since they were first developed. In 2007, Yang and Madden optimized the CPU cost of parameter tuning for ν-SVM [41]. They use particle swarm optimisation to calibrate the parameters. In 2010, a better decision function was improved by Tian and Gu based on the original function of Scholkopf [68]. As for SVDD, Luo et al., improved its accurateness by giving distinct costs to distinct system calls in 2007 [40]. In 2003, Mapping Convergence (MC) was presented by Yu et al. It is a one-class method that combines a weak classifier (as the first layer) and a SVM based classifier (as the second layer) [77]. The first layer is for extracting strong outliers that are very far from the positive class boundary. This gives MC better accuratenesses than other OSVM methods [77]. In next section, we will introduce the mathematical expressions of one of these two algorithms, ν-SVM.

Besides OSVM, there are also one-class methods based on other algorithms. In 2000, Manevitz and Yousef suggested a one-class neural network model, that filters documents by the objects from the "target class" only [43]. The problem with this neural network model is that it is sensitive with representation choices. In the same year, Letouzey et al. published a one-class decision tree [37]. But it is difficult to apply when the data has a high number of dimensions. The accuracy and robustness of this algorithm were improved by Li and Zhang in 2008 [38]. In 2001, Nearest Neighbours Description was proposed by Tax et al. It is a one-class k-nearest neighbours (kNN) algorithm, but with a defect that the model can become more sensitive to noise when increasing the amount of neighbours. One-class Bayes classifier was published much earlier. In 1997, Datta introduced a naive Bayes classifier with only samples from one class [17]. Wang also extent a one-class naive Bayes classifier in 2003 [71]. The newest one-class algorithm is the one-class logistic regression method suggested by Sokolov et al. in 2016 [63]. Other techniques include one-class anomaly detection, minimum spanning tree, and density estimator etc [62] [32] [26]. Although many special one-class methods were developed over the last 20 years, their authors still compare their performance with standard OSVM methods, since OSVM is still the most stable and applicable one-class algorithm [36].

OCC is not widely applied in bioinformatics research, although there are many medical data sets have unlabelled categories. Gardner et al. applied OCC on electroencephalography data in

2006 [23]. It shows the potential of combining OCC and biomedical data. OCC was also used to analysis multi-sequence magnetic resonance imaging (MRI) in 2015, in order to diagnosis multiple sclerosis lesions [34]. In 2016, Sokolov et al. presented the potential use of OCC on cancer diagnosis [63]. This is one the major study of applying OCC on gene expression data. They compared three main one-class machine learning methods, ν-SVM, SVDD, and one-class logistic regression [63]. The results show that ν-SVM and one-class logistic regression have similar performances on four microarray datasets, and they are always better than SVDD.

The literature review shows that there are not many applications about OCC on biomedical data, especially on gene expression data. Therefore, it is reasonable to evaluate the performance of OCC on gene expression data, in order to explore its possible use.

2.2 Mathematical Background

This section briefly introduces the basic concept of machine learning and mathematical terms used in this thesis. It is meant to explain these concepts to those reader without related backgrounds. It also gives the basic mathematical principles of the machine learning algorithms we use in this study: *binary SVM* and *one-class SVM*. Please notice that in this thesis, all vectors are denoted as bold letters, and the scalars are denoted as italic letters. For example, $\mathbf{x}$ denotes an vector, but x denotes a scalar.

2.2.1 Machine Learning

Machine learning, original proposed by Samuel in 1959, is an area of computer science that focuses on statistical models and algorithms [57]. It helps computers to do certain tasks without any specific programmed instructions. Instead, machine learning uses a certain algorithm to learn patterns and inferences from datasets. In his article, Samuel defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed" [57].

We can have an example to make this concept clear. Think about the voice assistant on your smart phone and why it can understand the word you said. If we consider your voice as a signal, there are certainly many features that we can extract from it: frequency, amplitude, or the time you spent to pronounce. For different sentences, for instance, the voice of saying “yes” or “no”, these features are also different. Now if we have a dataset containing 1000 persons speaking “yes” and 1000 “no”, and simplify the features as frequency, amplitude, and time length, we will have a 2000x3 matrix in a 3-d space. Since the 3 features are all different between the 2 groups, the data points from different class should cluster at different positions, which means we can use a 2-d plane to separate them. Then a new data point without any label can also be classified by which side it falls on. There are many mathematical algorithms that we can calculate what this 2-d plane is, and all these algorithms require input data. In our case, it is the 2000x3 matrix. The whole process above is called machine learning, and we call the 2000x3 matrix as “training set”. In reality, it is usually more complicated than our example here. A high-dimension dataset like 3000-d is also quite common. Therefore, many different algorithms are given or optimized by practitioners to handle different types of problems, such as Convolutional Neural Network, Support Vector Machine, Random Forest, etc.

As an independent subject, machine learning developed rapidly over the last 20 years. From the 1990s, related researchers started to borrow mathematical models from statistics and probability theories. Machine learning is also benefited by the development of Internet and computer hardware technologies. For example, we can use machine learning to analyse huge data sets nowadays because of the high-capacity disks and high-performance Central Processing Units (CPU).

Traditionally, machine learning tasks can be divided into two categories: *supervised learning* and *unsupervised learning*. In supervised learning, an algorithm needs labelled inputs (data) to build a mathematical model, then generates the outputs (class or value). According to the different outputs, we can divide supervised learning into classification and regression. When the outputs are discrete and limited labels, like our voice recognition model above, we call it classification. If the output is continuous, like predicting the temperature, we call it regression. Typical supervised learning methods include Support Vector Machine (SVM), Random Forests, Nearest Neighbours,

and Naive Bayes classifier.

In unsupervised learning, only unlabelled data is given to a machine learning algorithm. The algorithm need to find the structure of data, by grouping or clustering the data points. Unsupervised learning is usually used to discover the patterns in data. Typical unsupervised learning methods include Neural Network, Cluster Analysis, Anomaly Detection, etc.

Machine learning is used in many applications. Nowadays, shopping websites and online media use machine learning to learn the habits of users, then suggest commodities or programs that they are possibly interested. One of the most famous machine learning applications is "AlphaGo", an AI player developed by Google to play the board game Go . It used deep learning to learn how to play Go, and became famous after defeating many professional players. In 2017, it defeated Ke Jie, the number 1 ranked player in the world, which showed a huge potential use of deep learning in the field of artificial intelligence.

Although machine learning has achieved a huge success in various areas, it still has some limitations. One serious problem with machine learning is suffering from data biases. The outputs of machine learning models entirely depend on the training data inputted by human. Such bias can cause the inappropriate results, like racist and other presented unconscious biases [21]

2.2.2 Binary Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning method that can be applied both on regression and classification. The current standard version of SVM was invented by Vapnik and Cortes in 1995 [15]. Since then, it has been applied to many fields. The idea of SVM is characterized by its maximum margin property, or in other word, the decision boundary is chosen to be the one for which the margin is maximized [8].

The common two-class classification linear model can be described as follows:

$$y(x) = \mathbf{w}^T\mathbf{x} + b, \tag{2.1}$$

where $\mathbf{x}$ denotes an arbitrary vector in N-dimension space, b is the bias parameter, and the weights $\mathbf{w}$ are learned from the data. Therefore, we can get the corresponding decision boundary by following equation, $y(x) = 0$, which geometrically expressed as a $N-1$ dimensions hyperplane. In 2-D case, it is a line orthogonal to the weight vector $\mathbf{w}$. Then the classification rule can be written as:

$$F(x) = \begin{cases} \text{Class -1}, & \text{if } \mathbf{w}^T\mathbf{x} + b < 0; \\ \text{Class +1}, & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 0. \end{cases} \tag{2.2}$$

For a point x on the decision surface, $y(x) = 0$, according to the definition of dot product, we can have the distance from the origin to the decision surface:

$$\frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|} = -\frac{b}{\|\mathbf{w}\|}. \tag{2.3}$$

According to equation (2.3), we can see that the bias parameter b determines where the decision surface locates in the N-dimension space.

Then we can have distance from an arbitrary point x to the decision surface: $r = y(x)/\|\mathbf{w}\|$. This can be got when we draw a vertical line to the decision surface. Now we can denote its orthogonal projection point onto the decision surface as x_1. Since the vector from x to x_1 is parallel to the weight vector $\mathbf{w}$, according to the definition of vector addition, we can have:

$$\mathbf{x} = \mathbf{x_1} + r\frac{\mathbf{w}}{\|\mathbf{w}\|}. \tag{2.4}$$

It is easy to solve r by multiplying both sides of this equation by w^T and adding b. According to $y(x_1) = 0$ and (2.1), we can solve r as:

$$r = \frac{y(x)}{\|\mathbf{w}\|}. \tag{2.5}$$

It is common to write (2.1) as the following form when discussing SVM classifier:

$$y(x) = \mathbf{w}^T\phi(\mathbf{x}) + b, \tag{2.6}$$

where $\phi(\mathbf{x})$ denotes a fixed feature-space transformation, and the training data set are usually defined as N input vectors x_1, ... , x_N, with corresponding target values t_1, ... , t_N, where $t_n \in \{-1, 1\}$. The sign of $y(x)$ can define the class of a new data point. If we assume that the training data set is linearly separable in feature space, then there will be at least one set of parameter $(\mathbf{w}, b)$ exists. According to (2.2), for class -1, the parameter sets can give $t_n = +1$ for all data points that $y(x_n) > 0$, and $t_n = +1$ for those $y(x_n) < 0$, so that for all the data points, $t_n y(x_n) > 0$.

Often there are several parameter sets $(\mathbf{w}, b)$ exist, which means there could be several vectors in the feature space that can separate all the training points. Therefore, it is significant to find the best parameter sets that gives the smallest generalization error. From this perspective, the support vector machine is a method that solves this problem through the concept of the margin, which is defined to be the smallest distance between the decision boundary and any of the samples [8].

From (2.4), we have the formula of distance from an arbitrary point to the decision surface, where $y(x)$ can takes the form of (2.6). Furthermore, the solution should satisfy $t_n y(x_n) > 0$ so that it can classify all training points correctly. Therefore the distance can be given by following equation:

$$\frac{t_n y(x_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^T\phi(\mathbf{x}) + b)}{\|\mathbf{w}\|}. \tag{2.7}$$

Thus the maximum margin solution of parameters set can be found by solving:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n(\mathbf{w}^T\phi(\mathbf{x}) + b) \right] \right\}, \tag{2.8}$$

and it can be simplified into an equivalent form [15]:

$$\begin{aligned} &\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\| \\ &\text{subject to } t_n(\mathbf{w}^T\phi(\mathbf{x}) + b) \geq 1, n = \{1, 2, ..., N\}. \end{aligned} \tag{2.9}$$

The solution of 2.8 can be gained by quadratic programming algorithm and Lagrange multipliers, and there are many tailored implementations exist [8]. The implementation of SVM used in this study is based on the used LibSVM library, a popular open SVM library developed at the National Taiwan University [12].

For all the procedures above, we assumed that all training samples are classified, but in reality we have to allow some samples to reside on the wrong side of the margin. Therefore, the penalization is introduced. For this purpose, $\mathbf{w}, b : \mathbf{x} \rightarrow f(\mathbf{x}) = sgn(\mathbf{w} \cdot \phi(\mathbf{x}) - b)$ in chosen to minimize the resulting function.

The SVM solution can be extended to non-linear boundaries using the kernel trick. Essentially, it is a method that can map the data into a higher dimension, then design a linear SVM there. A basic kernel that maps 2D data into 3D can be described as $(x, y) \rightarrow (x, y, \sqrt{2})$. which is called explicit polynomial kernel. This kernel can transform 2D data into 3D explicitly and fit the SVM with transformed data, but it is slow to compute. In practice, implicit mapping is much more popular. In implicit mapping, it substitutes each dot product in the SVM algorithm by the kernel $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{y})^2$ and fits with original 2D data. This is the key to efficiency.

There are many kernel functions for SVM. In our study, following kernels are tested for the both binary and OCC classifiers, then the kernel with the best performance was selected. Here lists the kernels that used in this study:

1 **Radial Kernel:** $\kappa(\mathbf{x}, \mathbf{y}) = \exp\big(-\frac{\|\mathbf{x}-\mathbf{y}\|}{2\sigma^2}\big)$;

2 **Linear Kernel:** $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})$;

3 **Polynomial Kernel:** $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d$;

4 **Sigmoid Kernel:** $\kappa(\mathbf{x}, \mathbf{y}) = \tanh(a\mathbf{x} \cdot \mathbf{y} + b)$, with $a > 0$ and $b < 0$.

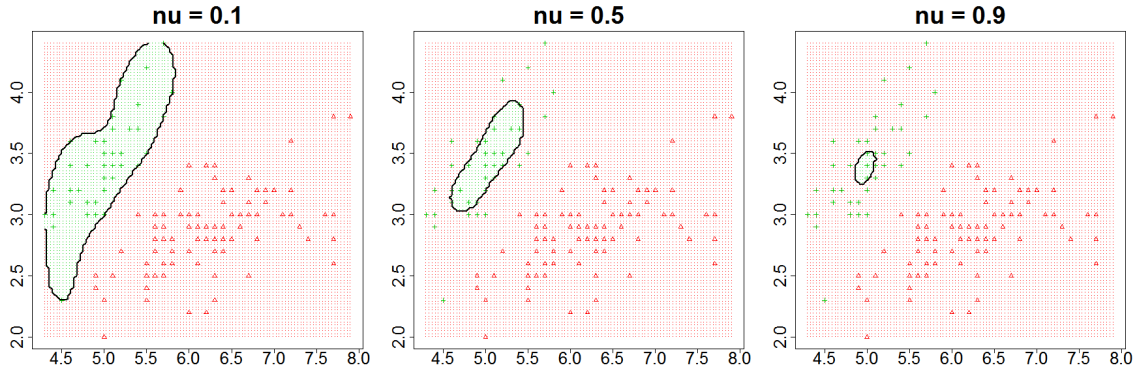

Figure 2.1. *The parameter ν affects how many samples are included by the decision boundary*

2.2.3 One-Class Support Vector Machine

ν-SVM is a one-class machine learning method proposed by Scholkopf et al [59]. Comparing with the binary SVM, we can get the determine function of ν-SVM, $f(\mathbf{x})$, by solving the following equation:

$$\begin{aligned} &\min_{\mathbf{w}\in F,\xi\in\mathbb{R}^l,b\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu l}\sum_i \xi_i - b \\ &\text{subject to } (\mathbf{w}\cdot\phi(\mathbf{x}_i)) \geq b - \xi_i,\ \xi_i \geq 0, \end{aligned} \tag{2.10}$$

where $\mathbf{x}_1, ..., \mathbf{x}_l \in \chi$ are the data points of training data, $l \in N$ is the observation times of training data, and χ is a compact subset of $\mathbb{R}^N$. Let ϕ be a feature map $\chi \to F$, i.e. a map into a dot product space F such that the dot product in the image of ϕ can be computed by evaluating a simple kernel, for example, radial kernel.

Like binary SVM, the solution of this algorithm can be gained by quadratic programming algorithm and Lagrange multipliers. It is shown in the original study of Scholkopf et al, so the specific proof process will not be explained in this thesis [59]. But it is still important to figure out one important parameter in this equation, ν.

When we have a new data point $\mathbf{x}$, we can determine the value $f(\mathbf{x})$ by judging which side of the hyperplane it falls on. Since non-zero slack variables ξ_i are penalized in the objective function, and if $\mathbf{w}$ and b are the solution of the equation above, the decision function $f(\mathbf{x}) = sgn(\mathbf{w}\cdot\phi(\mathbf{x}_i) - b)$ will be positive for the most training points $\mathbf{x}_i$, while the SV type regularization term $\|\mathbf{w}\|$ will still be small [59]. In this algorithm, the balance between these two goals is controlled by the parameter ν, furthermore, the parameter ν also characterizes the fractions of SVs and outliers [59]. Figure 2.1 shows the effect of the parameter ν. As we can see the greater the ν is, the less samples are included by the decision boundary. Therefore, it is the main parameter that we need to tune in this study.

2.2.4 Confusion Matrix and Balanced Accuracy

A confusion matrix is shown in Table 2.1. Traditionally, the term *accuracy* is defined as $(T^+ + F^+)/(T+F)$. It represents the ratio of correctly predicted samples and all samples. Two important statistical measures can be introduced here, *sensitivity* and *specificity*. Sensitivity (true positive rate) measures the proportion of correctly identified positive samples, denoted as $T^+/(T^+ + F^-)$. Specificity (true negative rate), by contrast, measures the proportion of correctly identified negative samples, denoted as $T^-/(T^- + F^+)$. The balanced accuracy (BAR) is defined as the average of sensitivity and specificity, which means it uses the information from the both sides [70].

Table 2.1. *Confusion matrix*

	Object from positive class	Object from negative class
Classified into positive class	True Positive, T^+	False Positive, T^-
Classified into negative class	False Negative, F^-	True Negative, F^+

2.2.5 Principal Components Analysis

Principal Components Analysis (PCA) was invented by Pearson in 1901 [53]. As the name suggests, it is a statistic method that can extract the main components from all the features. PCA applies eigendecomposition on the covariance matrix, then obtains the principal components (i.e. eigenvectors, abbreviated to PC) of the data and their weights (i.e. eigenvalues). It is meant to explain the variance of original data, or in other words, which direction of the data value has the greatest influence on the variance.

Generally, the results of principal components analyses are PCs. For the convenience of drawing plots, the typical number of PCs is two or three, though it can be technically more. As we introduced above, two PCs are two eigenvectors, which means we can draw a PCA plot by using PC1 as the x-axis and the PC2 as the y-axis. Figure 2.2 shows a sample PCA plot on an iris data set with two classes. There are four features in this dataset: sepal length, sepal width, petal length and petal width. We can see from the plot that these two classes are totally separated. The percentage after each PC represents how much variance this PC explains. If we add them up, the sum will represent the total percentage of variance that the PCA explains. If it is too low, the PCA may not explain the data set variance well. But if this percentage is high enough, it means what the plot shows is reliable.

PCA provides an effective way to reduce the data dimensions. It removes the components corresponding to the smallest eigenvalues in the original data, which means the low-dimensional data we get is optimized. In this study, we use PCA to evaluate our feature selection visually.

2.2.6 Basic Quartiles Terms and Box Plot

A box plot describes a group of numerical data by their quartiles and outliers. Before introducing how to read a box plot, let us introduce the definitions of quartiles in statistics. First of all, we need to sort the current number list in the order of their values, from the lowest to the highest. Then the "median" is the number in the middle of the sorted list. If there are two numbers in the middle, the median is their mean value. Since the lower quartile ($Q1$) and the upper quartile ($Q3$) have several definitions, we only introduce the definition we use in this study.

If the data size is an odd number, we can use the median to divide the whole data set into two parts. The lower quartile ($Q1$) is the median of the lower half, and the upper quartile ($Q3$) is the median of the higher half. If the data size is an even number, divide the data into two parts with the same size. Then $Q1$ and $Q3$ are the medians of these two parts. Interquartile Range (IQR), usually symbolized as ΔQ, is defined as $\Delta Q = Q3 - Q1$.

In a box plot, the upper bound of the data is defined as $Q3 + 1.5\Delta Q$, and the lower bound is defined as $Q1 - 1.5\Delta Q$. Any value greater than the upper bound or smaller than the lower bound is regarded as *outliers*. Figure 2.3 shows the box plots of the expressions of two genes. For each box plot, the top and bottom of the box are $Q3$ and $Q1$, and the band inside the box is the median. The whiskers represent the upper bound and the lower bound, and the outliers are plotted as small circles. We can see that the distributions between these two gene expressions are different. The $Q3$ of Gene 2 is even lower than the median of Gene 1. We can also know that the expression of Gene 2 has two outliers.

2.3 Biological Background

This section briefly introduces the basic biological terms used in this thesis. It is meant to explain them to readers without related backgrounds. It starts with the fundamental background

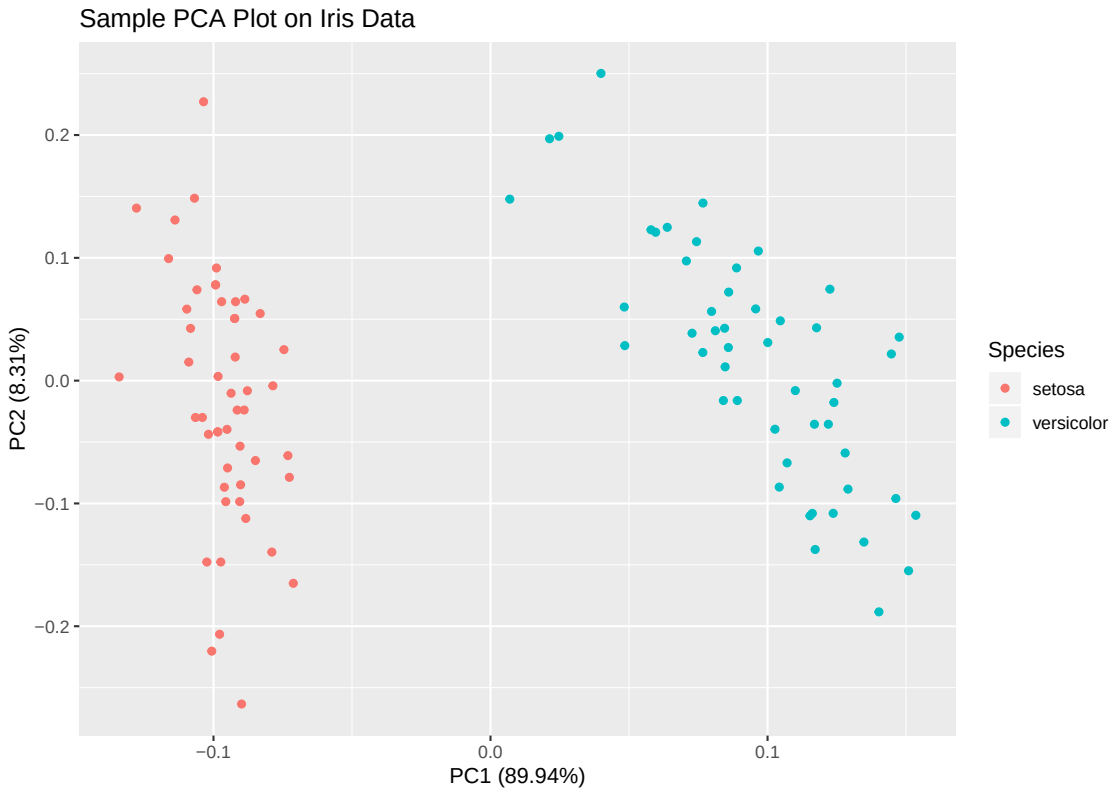

Figure 2.2. *A sample PCA plot on iris data. The plot shows that these two classes are totally separated. We can see from the legends that PC1 and PC2 explain 97.71% data variance in total. This is a very high percentage, which means we can trust the what the plot shows to us.*

of genetics: DNA, genes, gene expression, and what microarray data is. Then it explains the concept of cancer and the two diseases we studied.

2.3.1 DNA

The continuation of life depends on the hereditary information. It is passed from a cell to its daughter cell by the cell division, and passed to the next generation by its reproductive cells [2]. The most possible carrier element of hereditary information, deoxyribonucleic acid (DNA), was found in the 1940s. It stores these "genetic instructions" that determine the characteristics of species and each individuals [2].

The mechanism of DNA was not clear for a long time until its molecular structure was identified by Francis Crick and James Watson in 1953. Generally speaking, a DNA is two long polynucleotide chains made from repeating units called nucleotides [3]. The two chains bound to each other by hydrogen bonds and coiled around the same axis [73]. A DNA polymer can contain hundreds of millions of nucleotides, even though an individual nucleotide is quite small. After Crick and Watson, another possible carrier of hereditary information was found, ribonucleic acid (RNA). It carries the genetic information of many viruses. But since our study only focuses on Homo sapiens (or in more common words, human beings), and the genetic information of human is contained in DNA, we only introduce DNA and the related genetic processes in this chapter.

As the monomer units of DNA and RNA, nucleotides are organic molecules that have three distinctive chemical sub-units. Among them, the chemical sub-unit related to hereditary information is called *nitrogenous base*. In DNA, there are four bases found: adenine (A), cytosine (C), guanine (G), and thymine (T). Every single nucleotide has one of them. Nucleotides can pair with another on the other chain through their nitrogenous bases by the following rule: adenine - thymine and guanine - cytosine. This is usually symbolized as A-T and G-C base pairs [73]. The type of nucleotide is determined by its structure and nitrogenous base. Overall, there are thirty kinds of nucleotides. Fifteen of them belong to deoxyribonucleotide (unit of DNA), and the others

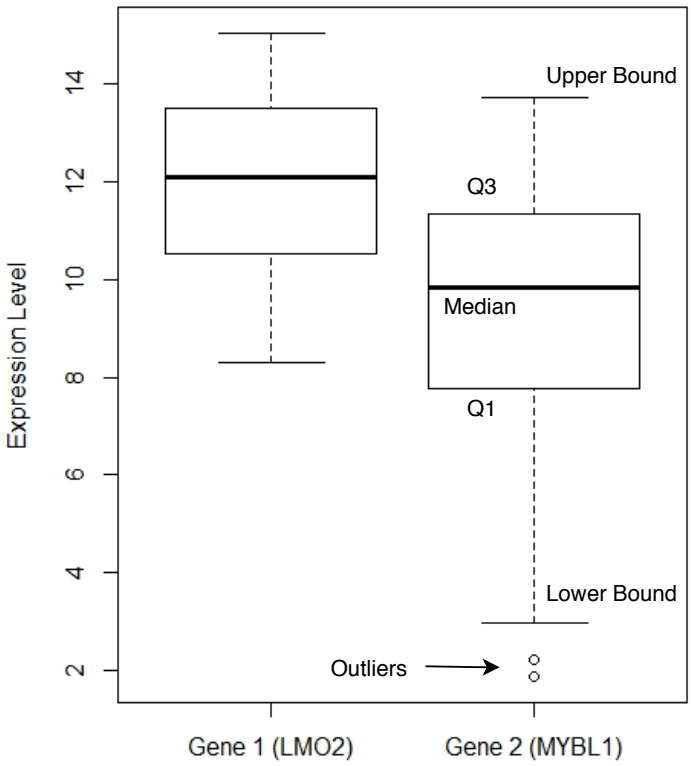

Figure 2.3. *Box plots of two genes expression. The top and bottom of the box are the higher quartile and the lower quartile, and the band inside the box is the median. The whiskers represent the upper bound and the lower bound. The outliers are plotted as small circles.*

belong to ribonucleotide (unit of RNA). The hereditary information is hidden in the sequences of these nucleotides. For instance, different creatures can different nucleotide sequences in their DNA. Figure 2.4 shows the idealized straightened out DNA molecular structure and the double helix structure.

2.3.2 Genes and Gene Expression

In addition to DNA, the term "gene" is also widely used when discussing hereditary information. Generally speaking, a gene is the piece of a DNA, that could be able to produce a certain protein. In other words, a gene is a sequence of nucleotides in DNA or RNA with a specific function. These sequences are called codons, which behave like the "words" in the genetic "language". In his study, Pennisi defines genes as "any discrete locus of heritable, genomic sequence which affect an organism's traits by being expressed as a functional product or by regulation of gene expression" [54].

A gene can have one or few names as its unique symbol to symbolize its function. These names are usually given by the HUGO Gene Nomenclature Committee (HGNC). In our study, all genes are described by their names.

The process of how genes affect human bodies is called "gene expression". It is a process that synthesizes functional gene products by hereditary information. In cells of Homo sapiens, this process is described by the "Central Dogma", which is usually stated as follows: "Transcription - Translation - Replication" or "DNA makes RNA and RNA makes protein" [16].

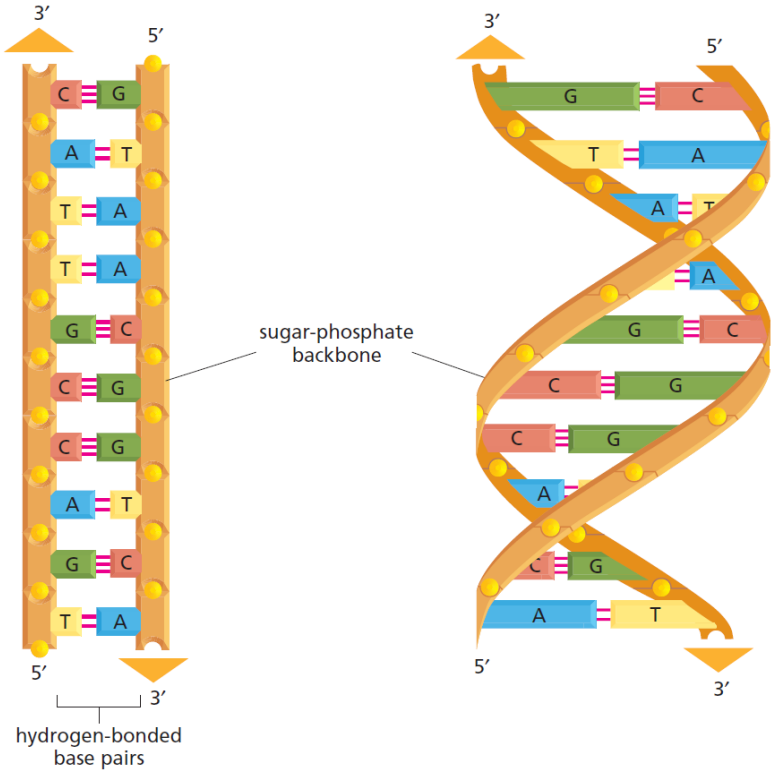

Figure 2.4. *The idealized straightened out DNA molecular structure is shown on the left, and the right part shows the double helix structure of DNA [2]*

The production of transcription is a RNA chain. It copies the hereditary information from the DNA. As a single chain of nucleotides, a RNA chain is synthesized by pairing nucleotides with DNA, just like one DNA chain pairs with another. The only difference is the thymines (T) in DNA are replaced with uracils (U) in RNA (symbolized as A-U and G-C). During the transcription, a gene on DNA will be read and copied to messenger RNA (mRNA). After editing and splicing, each mRNA will get close to ribosome to start the translation. The product of translation is protein. For Homo sapiens, it is the final product of gene expression.

Measuring gene expression products is quite important to modern medical study. For example, Zhao et al. has discovered that lymphoma patients with different lymphoma subtypes can have different expression level on certain genes [80]. Therefore, in order to classify the cancers subtypes, it is reasonable to use gene expression level to build the machine learning models.

2.3.3 Microarray Data

DNA microarray, firstly invented in 1983, is the most commonly used tool to measure gene expression product. Over the past 40 years, it is used to measure the expression level of a large number of genes [65]. As we mentioned in last subsection, the product of gene expression can be either mRNA or protein. However, microarray only measures one of them, which is mRNA. This is because the principle of microarray technique is based on the complementary property of nucleotides.

The principle of microarray technique can be generally stated as follows: the hybridization between one DNA strand and mRNA, or in other words, the complementary property of nitrogenous base pairs. In practice, a microarray chip is a solid surface with microscopic DNA spots on it. These DNA spots are called "probes". These probes can hybridize the mRNA in expression product sample, then generate fluorescent or electric signal. Microarray can compare the strength of these signals by using relative quantification and thereby measure the expression level of each gene.

Since the probes are small pieces of DNA, a probe can be gained by following ways: directly

	1007_s_at	1053_at	117_at	121_at	1255_g_at
GSM345077.CEL	6.446610	5.099553	5.280192	7.314208	2.728753
GSM345078.CEL	7.008966	4.911590	5.250850	7.274120	2.852769
GSM345079.CEL	7.030311	5.399787	5.029883	7.309508	2.508900
GSM345080.CEL	6.799201	5.537612	5.110150	7.406119	2.752950
GSM345081.CEL	7.581627	5.186826	5.145612	7.692765	2.823900
GSM345082.CEL	6.892375	5.097054	6.182779	7.481578	2.635250
GSM345083.CEL	7.280743	5.275250	4.848024	7.496912	2.469840
GSM345084.CEL	7.133561	5.162718	5.014665	7.380249	2.540985
GSM345085.CEL	7.001585	5.240952	5.609761	7.447812	2.558407
GSM345086.CEL	6.929950	4.996162	5.246502	7.206836	2.377894

Figure 2.5. *An example of microarray data in RStudio environment (RMA normalized)*

synthesizing a short sequence, or a reverse transcription from mRNA to cDNA. The second way utilizes the complementary property of nitrogenous bases we mentioned above (A-U, G-C).

In practice, a probe is shorter than a gene. For Affymetrix array, a famous microarray brand, each target gene is represented by a probe set that contains 11 to 20 probe-pairs [50]. Each probe pair is combined by one Perfect Match (PM) probe and one MisMatch (MM) probe. A PM is a probe that is perfectly complementary to the target, while a MM is a probe whose central base is mismatched [50]. Generally, MM probes are used to measure the amount of non-specific bindings. Often there are tens of thousands probe sets in a microarray dataset. Figure 2.5 shows a microarray dataset in RStudio environment. The row names are the sample ids on NCBI database, and the column names are the probe sets ids.

Because of the existence of MM probes, it is important to do normalization when analysing microarray data. One common method of normalization is Robust Multi-array Average (RMA) [28]. However, it only summarizes the perfect matches and mismatch spots are not utilized. The *Median Polish Algorithm* is used in RMA method of summarizing the information from perfect matches [24]. This is a robust exploratory data analysis procedure proposed by John Tukey in 1977. It it is meant to find an additively-fit model for two-way layout data by its row effect, column effect, and overall effect [48]. The general description of Median Polish Algorithm can be explained as follows:

1. Find the *overall effect*: put all medians of each row in a vector, and find the median of this vector;
2. Each element in the first row minuses the median of this row, then repeat this for each row;
3. Subtract the overall effect from each row median;
4. Do the same thing to each column, and add the overall effect from column to the add the overall effect that was received before;
5. Repeat steps 1 - 4 until very insignificant change occurs with row or column medians.

There are also other components in RMA normalization, such as quantile normalization, a method of normalizing a batch of arrays, and log-2 transformation. These procedures make the magnitudes of data at a comparable level for each the probe set and sample.

2.3.4 GEO and NCBI

GEO is the abbreviation of Gene Expression Omnibus, one of the largest public databases of high-throughput data. It also includes the information about related chips, microarray, RNA

sequences, and other similar data types [14]. It provides tools for users so that they can download and analysis the data. The GEO databases are supported by National Center for Biotechnology Information (NCBI).

2.3.5 Cancer

Cancer is the combination of a group of diseases, caused by abnormal cells that keep unstoppable divisions in the lesion. Normally, programmed cell death is a regular process in human bodies. Cells die when they get damaged or become old. In biological term, this is called apoptosis. However, the abnormal cells of cancer tumours survive even when they are old or damaged, and new cells are still generated when they are not needed. The abnormal cells can spread to surrounding organs and tissues, then result in their death. Finally, it leads to the death of patients. Cancer is the main reason why mortality happens on human beings. In 2012, cancer caused around 8 million death worldwide. Meanwhile, approximately 14 million new cases happened in the same year [64].

There are many risk factors that may cause cancers, such as tobacco use, obesity, ionizing radiation, environmental pollutants, and lacking physical activity. Alcohol intake was also find as a major cause of cancers [39]. Infections of microbes such as Helicobacter pylori, human papillomavirus infection, and human immunodeficiency virus (HIV) are also the main reasons why cancers happen, especially in developing countries. Beside above, an unavoidable risk factor of cancer is the genetic reason. It has been proved that all cancers happen with genome mutations [64]. It means that the encodes of some genes are permanently changed in the DNA sequence. Furthermore, some genes can differentially express between cancer patients and healthy people, which means it is potential to used gene expression data to do cancer classifications [1].

There are various of criterion to divide a cancer into subtypes. Clinically, it is common to do it by different treatment effects, e.g. the five-year survival rate. Five-year survival rate is one option to evaluate how the therapies work, especially on those aggressive diseases with a shorter life expectancy following, for example, the lung cancer. Two subtypes of one cancer may react differently to the same medicine, which may lead to different five-year survival rates. Nowadays, molecular subtypes are also often used on cancer classification because of the development of gene-targeted therapies. For example, there are two molecular subtypes of the basal bladder cancers, "epithelial" and "mesenchymal" [44]. Though the criterion of molecular subtypes still focuses on the survival rate, it gives more specific explanation in the aspect of molecular biology. Therefore, it is more popular in recent studies.

2.3.6 Breast Cancer

Breast cancer is a cancer that develops from breast tissue, accounting for 25% of all cancer cases [64]. It happens 100 times more often in women than men, which make itself the leading type of cancer in women worldwide. Usually, a patient will find a lump in the breast tissue or under armpits, that different from the other parts of the breast. Other symptoms includes the shape or position changing of the nipples, and the pain under armpits or around the collarbone [74]. Figure 2.6 shows the mammographies of a healthy breast (left) and a breast with breast cancer.

The risk factors include environmental and hereditary factors. Among them, the preventable environmental factors, such as alcohol intake, fat gain, lacking exercise, ionizing radiation, and chemicals, cause approximately 70% of the cases [42]. The other 30% of the cases are caused by the hereditary factors, which are usually non-preventable, for instance, genetic reasons. There are two high-risk genes identified, BRCA1 and BRCA2, that cause the most familial predisposition cases [64]. Besides, the expression of estrogen receptors (ER) and progesterone receptors (PR), and the over-expression of human epidermal growth factor receptor 2 (HER2) protein are also three main reasons why breast cancer occurs [6] [64]. Receptors are certain kinds of protein molecules that can receive chemical messengers, like hormones, from outside a cell. Breast cancer cells may have these receptors on their surface, cytoplasm or nucleus.

The biological reason why receptors can cause breast cancer is still not clear. In 2006, Deroo et al. proposed two hypotheses of the relationship between ER and breast cancer. The first hypothesis is that mutations happen when estrogen binds to ER. It stimulates the proliferation of

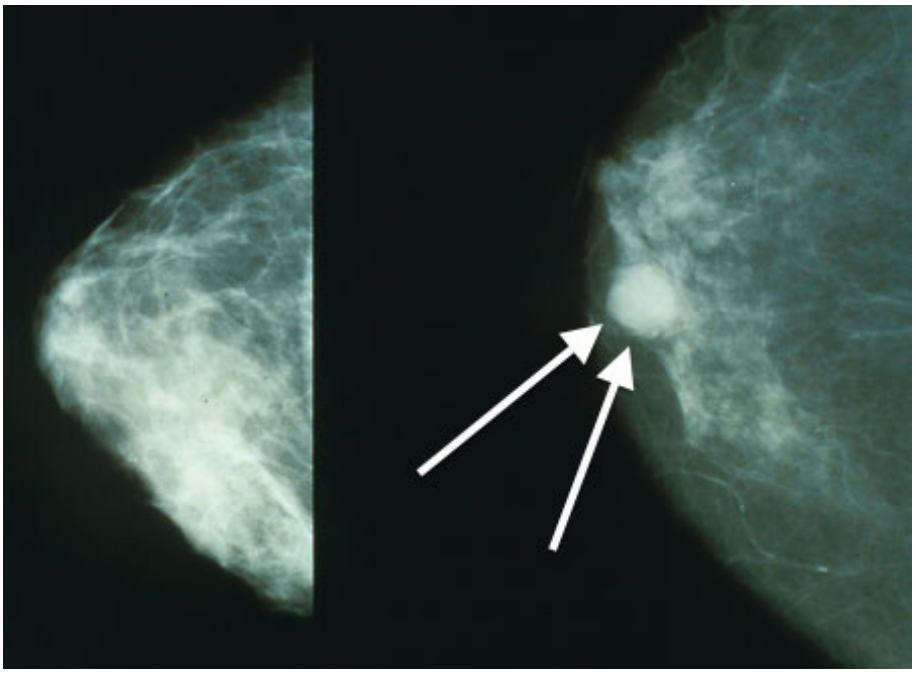

Figure 2.6. *Mammographies of a healthy person (left) and a breast cancer patient [5].*

mammary cells, since this process increases in cell division and DNA replication [18]. Moreover, estrogen metabolism can produce genotoxic waste, which may also increase the risk of breast cancer [18].

Clinically, it is very important to know if a receptor exists in the cancer cells. If breast cancer cells have ER receptors (usually called ER positive or ER+), the growth of these cells depends on estrogen. Then we can make the cancer cells necrosis by applying drugs that block estrogen effects. Therefore, it is reasonable to classify breast cancer by three following ways: ER positive or negative, PR positive or negative, and HER2 positive or negative. In our study, we divided breast caners into ER+ subtype and ER- subtype, then used them as the two classes of machine learning models.

2.3.7 Diffuse Large B-cell Lymphoma

Lymphoma is a kind of haematological malignancy, or generally speaking, a blood cancer. Worldwide, lymphoma has become the seventh most common form of cancer in the term of "cancer incidence" [64]. In 2012, there were around 566000 new cases of lymphoma and 305000 deaths caused [64]. Symptoms can be found on early stage patients, including enlarged lymph nodes without pain, fever, night sweats, weight loss, and tiredness.

Traditionally, lymphoma can be classified into two main categories: Hodgkin's lymphoma (HL) and non-Hodgkin lymphoma (NHL). Recently, the World Health Organization (WHO) added two new subtypes, which are multiple myeloma and immunoproliferative diseases [64]. As the name suggests, non-Hodgkin lymphoma (NHL) includes all types of lymphoma other than the Hodgkin's. Comparing with non-Hodgkin lymphoma, Hodgkin's lymphomas patients can have much better treatment effects. For those patients who are under 20-year old, the 5-year survival rates are around 97% [31]. However, for non-Hodgkin's lymphoma patients, it is only around 71% [60]. In our study, we use the data of *Diffuse Large B-cell Lymphoma (DLBCL)*, a subtype of non-Hodgkin lymphomas. It is the most common subtype that accounts for 40% of lymphoma cases worldwide [64]. Figure 2.7 shows the micrograph of diffuse large B cell lymphoma.

The causes of DLBCL are still not clear, but there are some hypotheses, such as underlying immunodeficiency and the infection of Epstein–Barr virus (EBV) [52]. Genetically, it is found by Morin et al., that genes with candidate mutations are ubiquitous in DLBCL patients [47]. Furthermore, a recent genome-wide association study of B cell non-Hodgkin lymphoma reveals that 3q27 is identified as a susceptibility locus in the Chinese population [33].

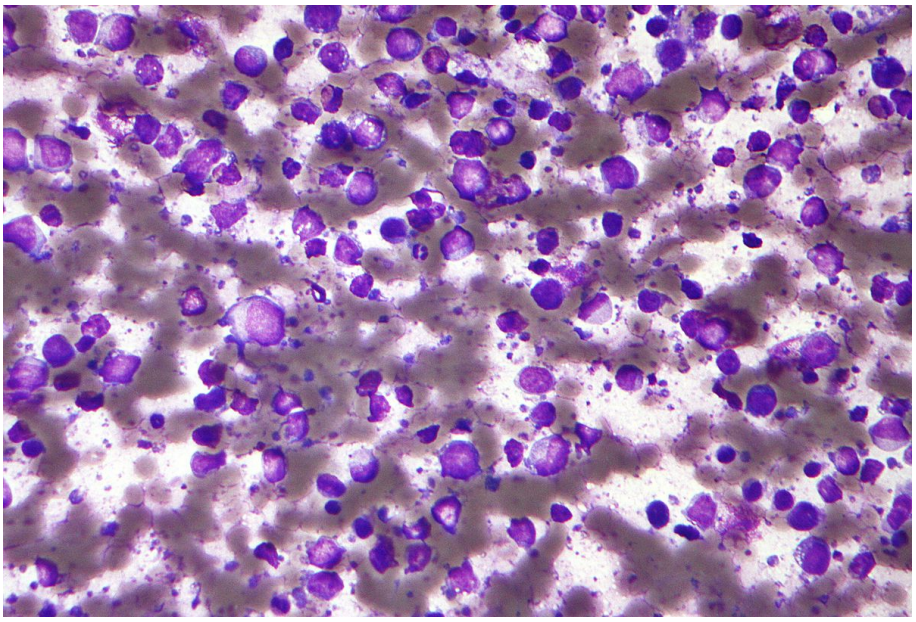

Figure 2.7. *Micrograph of a diffuse large B cell lymphoma [51].*

In 2012, Care et al divided DLBCL into three subtypes: germinal centre B-cell like (GCB), activated B- cell like (ABC), and unclassified (UC) subtype [9]. It is reported by Tirado et al that GCB patients have a much higher 5-year survive rate (59%) than ABC patients(30%) [69]. Moreover, the study of Zhao et al showed that there are eight significant genes that are related to the subtype formation [80]. Based on these studies, we think it is potential to build a classification machine learning model of DLBCL subtypes.

It is worth to mention that the unclassified subtype is not an independent subtype. Instead, it includes cases between ABC and GCB that cannot be clearly classified [80]. Clinically, it is much important to find those patients who have more dangerous subtype, in order to give them special treatments. In our case, it means we can regard GCB as the "safe" subtype, and all non-GCB subtypes as the "dangerous" subtype. Therefore, we divide all the patients into two groups, GCB and non-GCB, then build classification models based on these two classes.

3 MATERIALS AND METHODS

In this study, we evaluated two algorithms, ν-SVM and binary SVM. This chapter will introduce the materials and the R applications we use, how we do the data pre-processing, and the overall workflow.

3.1 Algorithm Applications

In Chapter 2, we saw that OSVM is much more applicable and stable than other one-class methods. Hence, we did not choose any one-class methods other than OSVM. In this study, two OSVM methods were considered at the beginning, ν-SVM and SVDD [66] [59]. But as we mentioned in Chapter 2, ν-SVM achieves much better results than SVDD in the research of Sokolov et al [63]. Thus, we select ν-SVM classifier as the OCC machine learning method, and compare its performance with the standard binary SVM classifier.

Because OCC classifiers are commonly used on imbalanced data, it is inappropriate to use "accuracy" as the criterion to evaluate an algorithm. In imbalanced data, one class often has much more samples than the other. A classier may achieve very high performance when identifying the major class, but has very low ability to detect the minority class samples. Therefore, in this case, it is still not a proper classifier to detect the outliers. In order to solve this problem, Villalba et al. introduced *balanced accuracy (BAR)* to evaluate OCC algorithms [70]. The BAR is defined as the average of the true positive rate (sensitivity) and true negative rate (specificity) [70]. For more details about the definition of BAR, see Chapter 2. In this study, we use BAR as the criterion of all machine leaning models.

The machine learning application we use is the R Package "e1071", developed by Meyer et al. It focuses on short time Fourier transform, fuzzy clustering, support vector machines, and other statistics methods [45]. We use the "svm" function of this package to build the machine learning models. The one-class SVM algorithm in this package is the ν-SVM as we described in Chapter 2, and the binary SVM is the standard SVM algorithm. The evaluation function "ConfusionMatrix" is a built-in function of the package "caret" [30]. It gives the confusion matrix by the prediction and the test labels. We can get the BAR directly from the function output.

3.2 Training Data and Validation Data

All data sets we use in this study are microarray data. As we mentioned in Chapter 2, microarray contains the information of gene expression level. The changing of gene expression level may indicate a disease happens, or the difference between subtypes [25]. The normalized data of the two cancers are directly downloaded from Gene Expression Omnibus (GEO) database.

For the breast cancer experiment, a group of 200 patients from GSE11121 is treated as training set and a group of 198 patients from GSE7390 is the test set [58]. Each data set is divided into the two groups we introduced in Chapter 2: ER positive and ER negative. Comparing to ER negative patients, an ER positive patient usually has a better survival prognoses [13]. In the experiment of lymphoma, a group of 414 patients from GSE10846 are treated as training set and a group of 118 patients from GSE53786 are as the test set. The original data has three subtype labels: GCB, ABC, and unclassified (UC) [80]. As we see in Chapter 2, GCB lymphoma patients has really high 5-year survival rates comparing with other subtypes [69]. Therefore, we combine ABC and UC subtypes as the "Non-GCB" subtype, and the data is manually labelled into two groups: GCB and Non-GCB.

As described above, instead of doing cross-validation, we test the machine learning models

Table 3.1. *Data sets used in this study.*

Disease	Usage	Data Set	Neg.	Neg. PCT	Pos.	Pos. PCT	Samples
Breast Cancer	Training set	GSE11121	38	19%	162	81%	200
	Test set	GSE7390	64	32%	134	68%	198
Lymphoma	Training set	GSE10846	183	44%	231	56%	414
	Test set	GSE53786	45	38%	73	62%	118

on external data sets. The main reason why we did not choose cross-validation is the effect of the instrumentation or procedures. Experimenters may make mistakes when doing the experiment, and these mistakes may repeat during the whole experiment. The environment of laboratories can also affect the results, for instance, the temperature. Using an independent data set as the test set can avoid the interference above, because different contributors can hardly make the same mistake in the experiments.

Table 3.1 lists the data sets we use. In this study, we call the class with more sample points as the *major class*, and the other class as the *minority class*. In order to make OCC classifier to have better performance, we use the major class as the OCC training set. The major class is labelled as positive or 1, while the minority class is labelled as negative or 0. In breast cancer data sets, the positive class is the ER+ subtype. In lymphoma datasets, it is the Non-GCB subtype. We use the entire training sets to train the binary model, but only the positive class is used to train the OCC models.

3.3 Data Pre-processing

As described in Chapter 2, the downloaded microarray data is with probe ids as its rows, and sample ids as its columns. Therefore, our first step is converting probe ids to gene names, then we can select the gene we want. It is worth to mention that one probe may point at several genes in microarray data. Since these probes cannot give the expression level of a certain gene, we cannot use them as features. Therefore, we delete these probes before analysing. The annotation from probes to genes can be done by the annotation function. This is provided by the microarray platforms. In our study, the data sets are generated by following microarray platforms: [HG-U133A] Affymetrix Human Genome U133A Array and [HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array [10] [11]. So we download the corresponding packages, "hgu133a.db" and "hgu133plus2.db", to apply the annotations. The data sets are also transposed to suit the requirement of the machine learning application. The following R commands can annotate probe ids to gene names:

```
probe = colnames(microarray_data)
gene_table = AnnotationDbi::select(hgu133plus2.db,keys=probe,columns=c("SYMBOL"))
```

As a major input, the variable "columns" contains the column names of the transposed data, which are the probe ids. The code returns a gene table. Figure 3.1 shows a truncated gene table in R studio environment, whose first column, "PROBEID", contains probe ids, and the second, "SYMBOL", contains the gene names. If one probe points at several genes, the "SYMBOL" of this probe id will be empty. In this study, they are deleted before analysing.

Moreover, a microarray dataset may contain several probes that map the same gene. A common way to compare them is to keep the probe with the highest median value, and remove the other [72]. Since there are around 20000 probes in a dataset, we write a small function to clean the replicate probes automatically. Program 3.1 shows the R code of this function, where the data must be a transposed microarray matrix.

After annotation, the data is labelled into positive or negative groups by their metadata. Since the downloaded data is already normalized by RMA method, no more normalization is needed.

	PROBEID	SYMBOL
1	220951_s_at	A1CF
2	217757_at	A2M
3	219488_at	A4GALT
4	221131_at	A4GNT
5	218075_at	AAAS
6	218434_s_at	AACS
7	205969_at	AADAC
8	202851_at	AAGAB
9	202852_s_at	AAGAB
10	205434_s_at	AAK1

Figure 3.1. *A truncated gene table in R studio environment*

```
####### compare different probes with the same gene #######
# sort the gene_table by gene name before this function
gene_median_comparison <- function(data, gene_table){
  maxium = 1
  marker = matrix(0,nrow(gene_table), 1)
  marker[maxium] = 1
  i = 2
  for (i in 2:nrow(gene_table)) {
    if(gene_table$SYMBOL[i] == gene_table$SYMBOL[i-1]){
      if(median(data[,gene_table$PROBEID[i]])
         > median(data[,gene_table$PROBEID[maxium]])){
        marker[i] = 1
        marker[maxium] = 0
        maxium = i
      }else{
        next()
      }
    }else{
      maxium = i
      marker[maxium] = 1
    }
  }
  gene_table = cbind(marker, gene_table)
  gene_table = gene_table[gene_table$marker == 1, ]
  return(gene_table)
}
```

Program 3.1. *The R function that removes all low-expressed probes with replicated use*

3.4 Feature Selection

Some studies have indicated that breast cancer and DLBCL are highly associated with the expression of certain genes. Therefore, we select features based on these studies. It is reported by Wirapati et al., that the following seven genes are related to the formation of breast cancer tumours: AURKA, PLAU, STAT1, VEGF, CASP3, ESR1, and ERBB2 [75]. Particularly, ESR1 represents the ER signalling. Therefore, all the genes above are used to build the machine learning model. As for DLBCL, Zhao et al. introduce that the following eight genes are related to the subtype conformation: MYBL1, LMO2, BCL6, MME, IRF4, NFKBIZ, PDE4B, and SLA. These genes express differently in the patients of Non-GCB and GCB subtypes, which made themselves expectable features [80]. In order to see if potentiality of these features, we apply *Principal Component Analysis (PCA)* on all the datasets after feature selection.

3.5 Work flow

This section briefly explains the workflow after PCA. The data of each disease will be evaluated by the following procedures.

First, since there are four kernels for both binary and OCC classifier, we need to compare the performance of each kernel. For both diseases, we tuned parameters for each kernel by a 10-fold cross-validation in the training set. Then we select the best parameters of each kernel, build models with them, and test these models on the test sets. The results are compared with their BAR. Furthermore, since the parameter tuning may cause over-fitting, we reversed the training and test sets, then did the tests again. The workflow of reversed tests are the same as the normal tests.

There are some studies that indicate the performance of OSVM is better than binary SVM when only few negative samples exist [36]. Therefore, we applied imbalanced tests to see if OCC works better when we remove some negative samples. For each disease, we remove certain proportion of negative samples in the training sets in the following order: 25%, 50%, 75%, and 90%. Before each removal, the negative samples in the training set are sorted into a random order to keep the results more stable. Then we use the rest of the training data to build the binary models. For each percentage, the operation above is repeated for 10 times.

See Figure 3.2 for the overall work flow of this study. The replicated operations, like removing negative samples, are drawn only once.

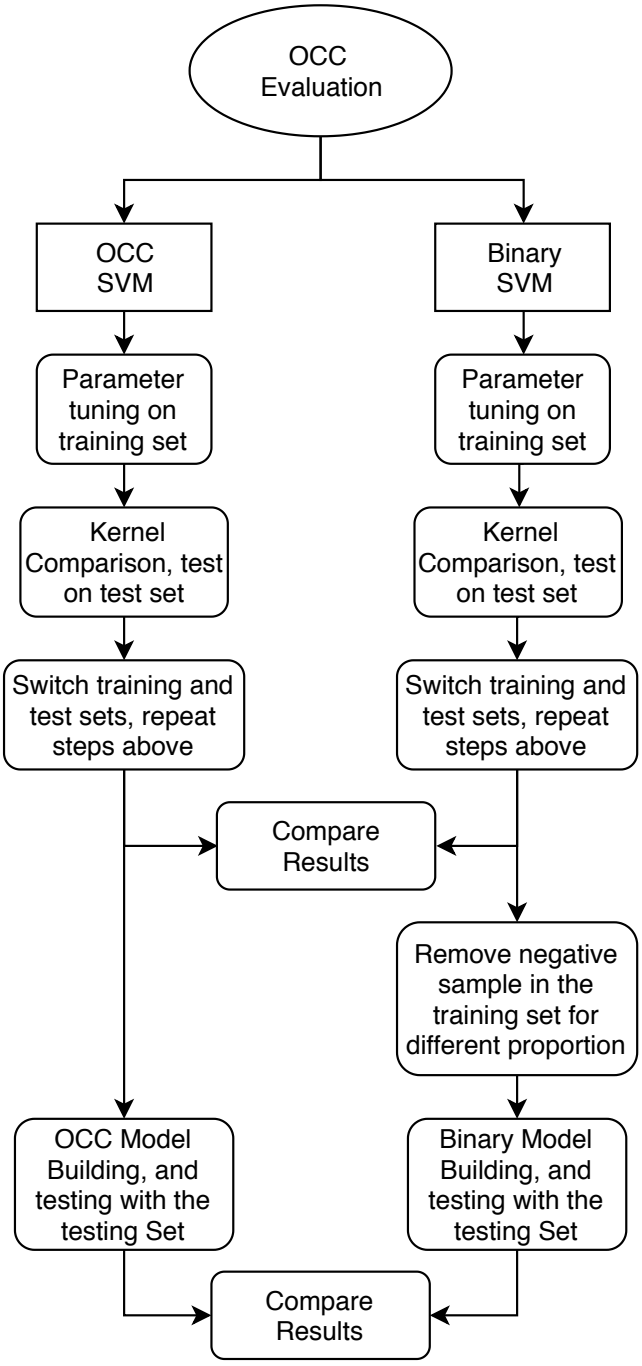

Figure 3.2. *The overall workflow after the principal component analysis*

4 RESULTS AND DISCUSSION

4.1 Results

After all the pre-processing steps, such as labelling, annotation, and feature selection, the datasets are checked by PCA. The purpose of PCA is to see if the features are selected successfully. Figures 4.1 and 4.2 show the PCA plots of all the data sets we use.

We can see from Figure 4.1 that the two classes of breast cancer are generally separable, both in the training and test set. Thus, it is possible to use these features to classify the ER+ and ER- subtypes. This result verifies the biological meaning of our predefined classes, that is, these genes differentially expressed between ER+ and ER- breast cancer patients. As for lymphoma data, we plot their PCA by their original labels, which are GCB, ABC and Unclassified. As we presented in Chapter 2, GCB patients have much higher 5-year survive rates after treatments, so it is better to divide the dataset into GCB (relatively safe) and non-GCB subsets (the combination of ABC and unclassified class, possibly dangerous). The PCA plots of lymphoma datasets support our division in another aspect. In Figure 4.2, we can see that the unclassified data points are between GCB and ABC. This supports the theory we mentioned in Chapter 2, that is, the unclassified subtype is not an independent subtype, but some unclear patients between GCB and ABC subtypes. Therefore, it is reasonable to combine ABC and unclassified classes as a single class, then compare it with GCB group. Furthermore, the PCA plots show that our feature selection of lymphoma data is successful since the classes are separated in these plots.

The machine learning models in this study are built in R environment. In Figure 2.1, we showed that the parameter ν has a huge impact on OCC SVM boundaries. Another important parameter, γ, decides how many support vectors there are. The larger our γ is, the fewer support vectors we have. A huge γ may cause the models only work surrounding the training points, so it is also an important parameter for both OCC and binary SVM. Therefore, it is necessary to tune these parameters before each SVM model builds. Appendix A shows our parameter tuning function in R codes. A parameter set of ν ($0 < \nu < 1$) is given to the function, and a parameter set of γ, too. In order to avoid using information from the test sets, the parameter tuning is done by a 10-fold cross-validation within the ***training set***. The polynomial kernel is tuned independently because it needs an additional parameter set, *degree*. Since we compare all kernels and many possible parameter values are tested, there are too many tuning results generated. Therefore, we only show the parameter combinations that give the best tuning results in Table 4.1 and 4.2. They also give the corresponding balanced accuracies. After comparing, the best kernel for breast cancer models is the sigmoid kernel, and the best kernel for lymphoma models is the radial kernel.

In Figure 3.2 and Chapter 2 we showed that it is important to select the best kernel for each classifier. In this study, all major kernels are compared: radial, liner, polynomial, and sigmoid. Since we have already gotten the best parameter sets for each kernel from parameter tuning, we can now build a model for each kernel by using the corresponding parameters. The results of kernel selection are shown in Figure 4.3 and Figure 4.4.

In Figure 4.3, we observe that the sigmoid and radial kernels have much higher performances than the other two kernels. Moreover, the balanced accuracies of these two kernels are both over 0.8. These results show expectable prospects of the classification. The kernel comparison in the lymphoma data sets return similar results, but only the radial kernel gives a high BAR for OCC-SVM. The specific BARs are listed in Table 4.3.

We can see from the plots that the best kernel for breast cancer models is the sigmoid kernel, while the best for lymphoma models is the radial kernel. Remind that we get these results on the external test sets, and they are the same as the best kernels we get from parameter tuning (cross-validation). For above reasons, we only use these two kernels in the following imbalance study.

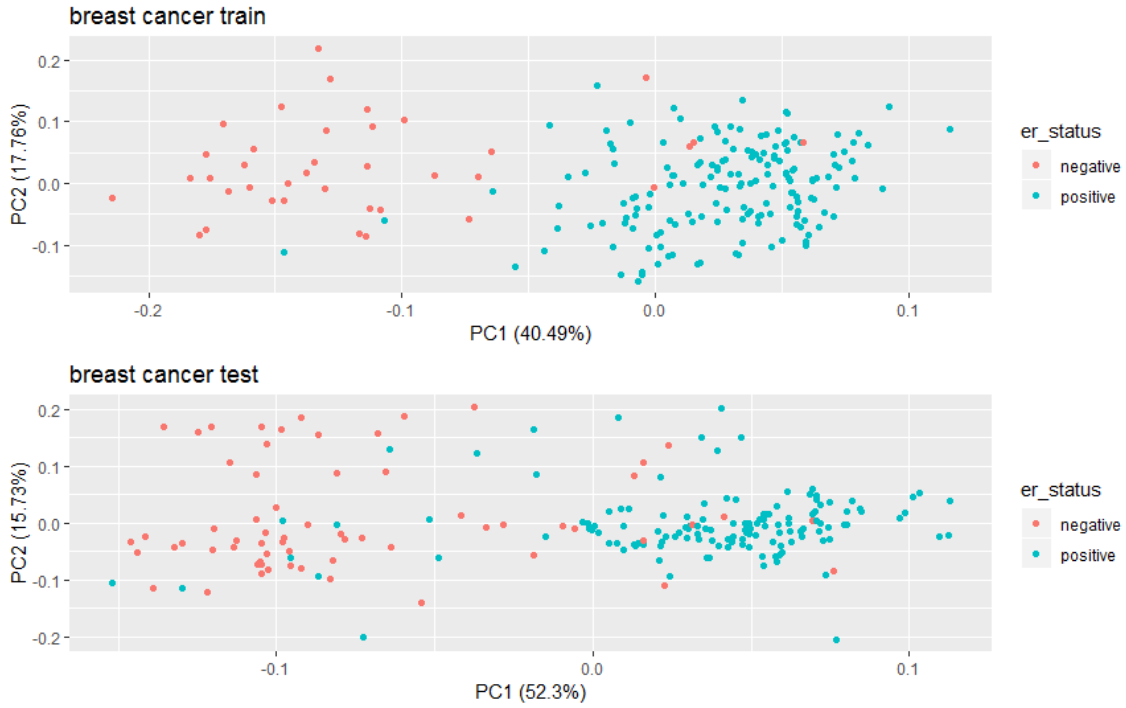

Figure 4.1. *PCA plots for breast cancer data sets. The blue points represent estrogen receptor(ER) positive samples, and the red points stand for ER negative samples. It is clear that the two classes clustered on different parts of the plots, so that it is potential to use our features to build a machine learning model.*

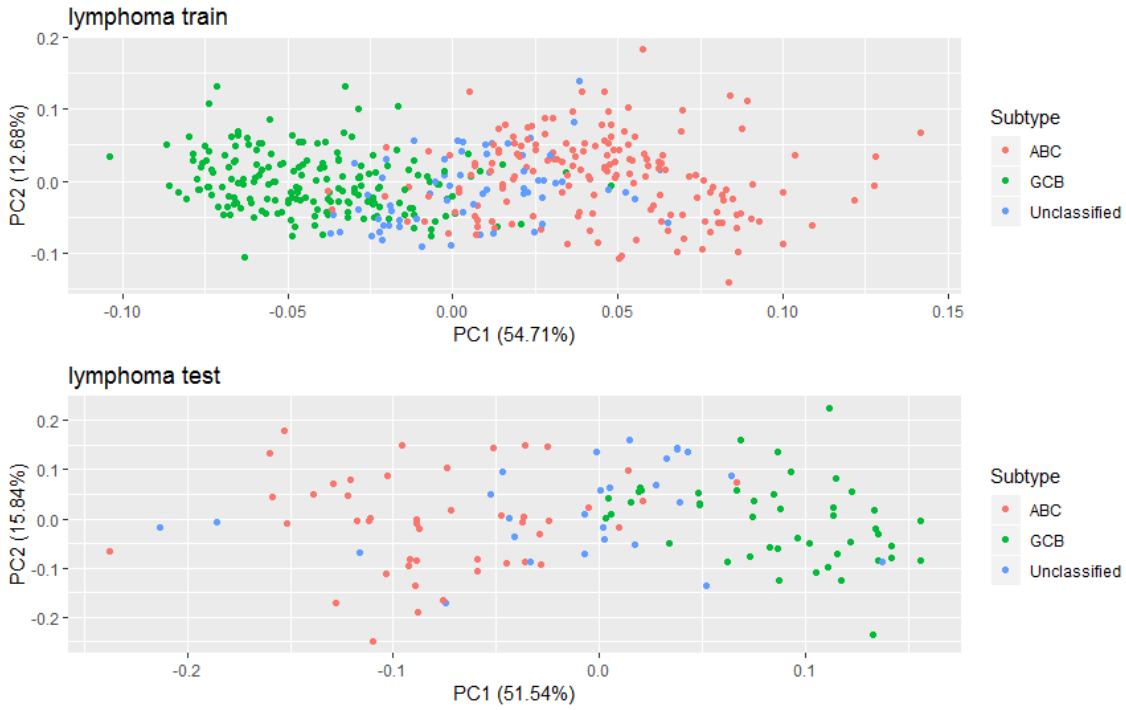

Figure 4.2. *PCA plots for lymphoma data sets. The red points represent ABC subtype samples, the green points stand for GCB subtype samples, and the blue points stand for unclassified samples. We can see from the figure that the ABC and GCB subtype clustered on different parts of the plots. Also the unclassified samples clustered in between with ABC and GCB, which explained why we can combine it with ABC as the "Non-GCB" subtype. It shows that it is potential to use our features to build a classification model.*

Table 4.1. *This table shows the best parameters for the* ***breast cancer*** *data and the corresponding balanced accuracies. The input vector of parameter ν is {0.005, 0.1, 0.2, ..., 0.9, 0.995}. The γ set is 1 / (1, 2, ..., 1000). For each kernel, we did a 10-fold cross-validation in the training set, then used the mean value of the BARs as the performance. Here we only show the parameter combinations that give the best tuning results. We can see from the table that the best kernel for breast cancer data is the sigmoid kernel.*

Classifier	Outputs	Linear	Radial	Polynomial	Sigmoid
OCC	Performance	0.640	0.873	0.531	0.890
	ν	0.5	0.005	0.1	0.2
	γ	N/A	1/32	1/7	1/25
	degree	N/A	N/A	3	N/A
Binary	Performance	0.926	0.923	0.868	0.933
	γ	N/A	1/15	1/8	1/9
	degree	N/A	N/A	3	N/A

Table 4.2. *This table shows the best parameters for the* ***lymphoma*** *data and the corresponding balanced accuracies. The input vector of parameter ν is {0.005, 0.1, 0.2, ..., 0.9, 0.995}. The γ set is 1 / (1, 2, ..., 1000). For each kernel, we did a 10-fold cross-validation in the training set, then used the mean value of the BARs as the performance. Here we only show the parameter combinations that give the best tuning results. We can see from the table that the best kernel for lymphoma data is the radial kernel.*

Classifier	Outputs	Linear	Radial	Polynomial	Sigmoid
OCC	Performance	0.573	0.792	0.624	0.531
	ν	0.8	0.1	0.005	0.1
	γ	N/A	1/26	1	1/49
	degree	N/A	N/A	3	N/A
Binary	Performance	0.901	0.910	0.890	0.902
	γ	N/A	1/26	1/4	1/12
	degree	N/A	N/A	3	N/A

Table 4.3. *The balanced accuracy of each kernel on the* ***test sets****. The comparison shows that the best kernel for breast cancer data is sigmoid kernel, and for lymphoma, it is radial kernel. On the same disease data, the best kernels are the same for both OCC SVM and binary SVM.*

Disease	Classifier	Radial	Linear	Polynomial	Sigmoid
Breast Cancer	OCC	0.818	0.389	0.590	0.859
	Binary	0.873	0.861	0.875	0.876
Lymphoma	OCC	0.881	0.686	0.754	0.678
	Binary	0.983	0.915	0.974	0.923

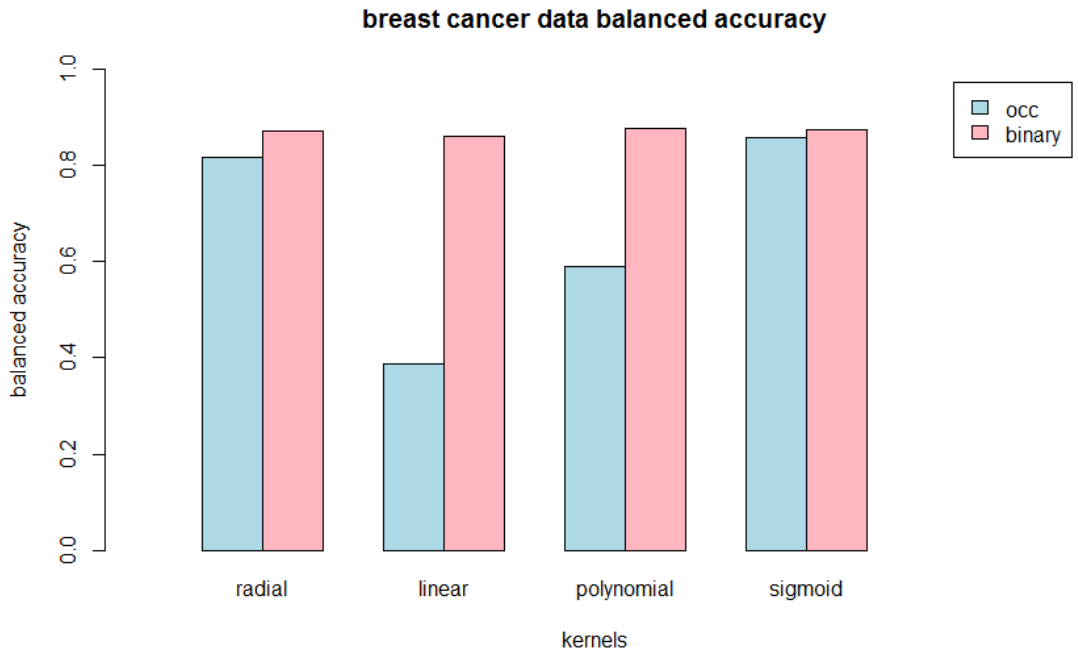

Figure 4.3. *Kernel selection results on the breast cancer* ***test set****. The blue bars represent OCC performances, and the pink bars represent for the binary classifier. The performances are evaluated by balanced accuracy. The binary classifiers with different kernels achieve similar performances. But only sigmoid and radial kernels give a good performance for the OCC classifier (BAR > 80%). The figure shows that sigmoid kernel is the best kernel on breast cancer data for both classifiers, OCC and binary.*

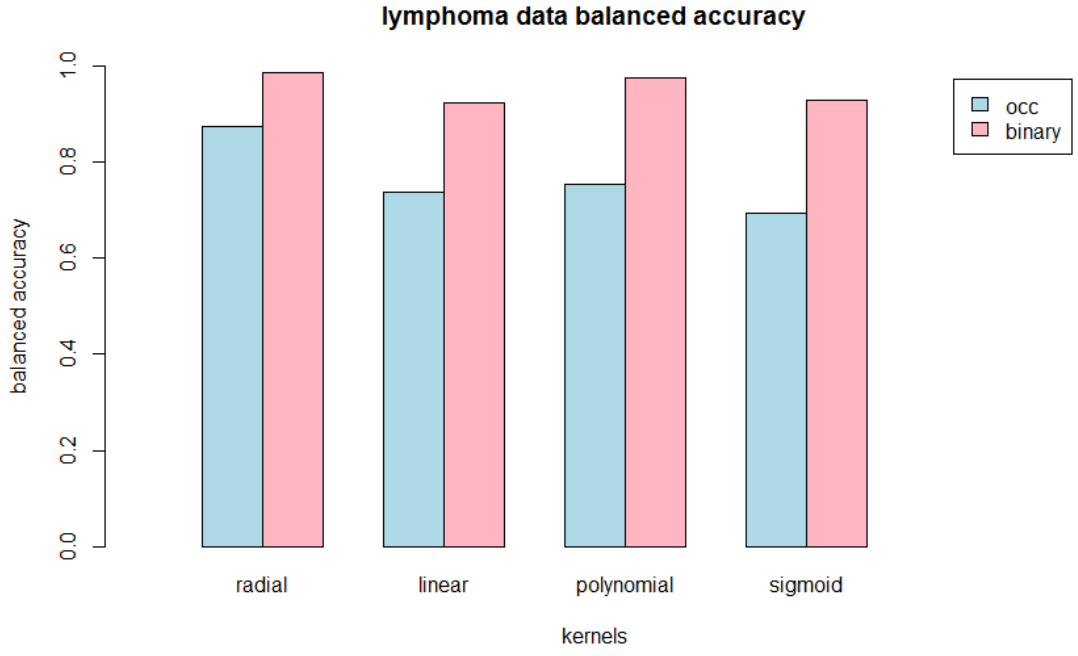

Figure 4.4. *Kernel selection results on the lymphoma* ***test set****. The blue bars represent OCC performance, and the pink bars represent for the binary classifier. The performances are evaluated by balanced accuracy. The figure shows that Radial kernel is the best kernel on lymphoma data for both OCC and binary classifiers.*

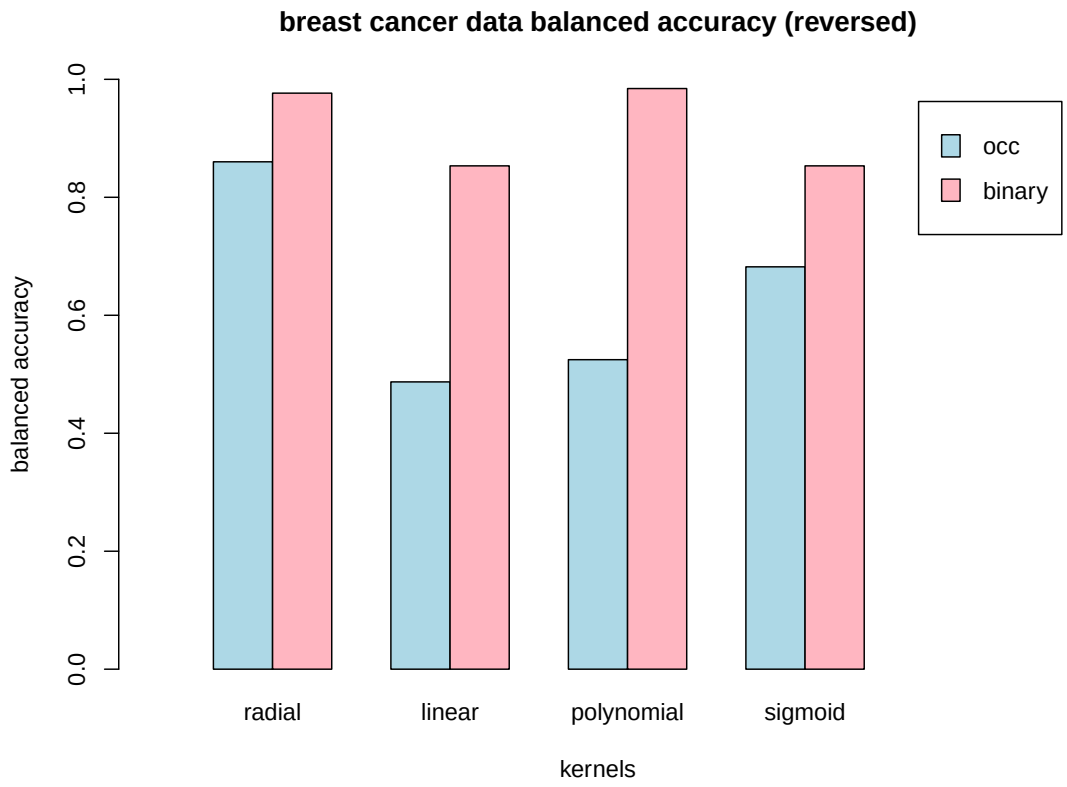

Figure 4.5. *Kernel selection results on the breast cancer data (trained with the test set and test on the training set). The blue bars represent OCC performances, and the pink bars represent for the binary classifier. The performances are evaluated by balanced accuracy. The best BAR of OCC is around 86% (Sigmoid kernel), and the best BAR of binary SVM is 97%.*

Furthermore, as stated in Chapter 3, we also reversed the training and test sets to avoid over-fitting. After tuning parameters on the test sets, we trained OCC and binary models with the test sets, then test these models on the training sets. These results are shown in Figure 4.5 and Figure 4.6. Both OCC and binary classifiers can achieve very high BARs after the reversing. These results support our previous findings, and reduce the possibility of over-fitting.

The four bar plots above show that binary models can achieve better performances than OCC on these data sets. Even though we select the best kernel for each classifier, the binary models still achieve higher BARs. But it is still too soon to say that binary SVM classifier is a better choice. First, OCC-SVM uses less memory resources since it only uses the positive class to train the model. In some cases, one may want to stand the reduction of performance for a better training speed. Especially in our case, the performance has no huge disparity between OCC and binary. Secondly, like Khan shows in his review article, binary classifiers may not work well when the classes are severely imbalanced [36]. In clinical situations, imbalance is not rare, like the example about rare cancer subtype we introduced in Chapter 1. Let us introduce a term called **class ratio**. It is defined as the ratio of the number of positive samples to the number of negative samples. The class ratios in our training sets are 81:19 (breast cancer) and 56:44 (lymphoma). Although the two classes of breast cancer data are somewhat imbalanced, they are still not *severely* imbalanced. As for lymphoma data, the two classes are basically *in balance*. Therefore, we did the following imbalance tests to simulate the imbalanced situations, then compare the performances between these two classifiers.

As we stated in Chapter 3, we used all the training samples to build the binary models, but only the positive samples are used to build the OCC models. Now we try to reduce the ratio of negative samples (the minority class) of the binary training sets. We remove 25%, 50%, 75% and 90 % of negative samples respectively from the training sets. These removed samples are selected randomly. The training sets of OCC-SVM remain fixed since there is no negative sample within

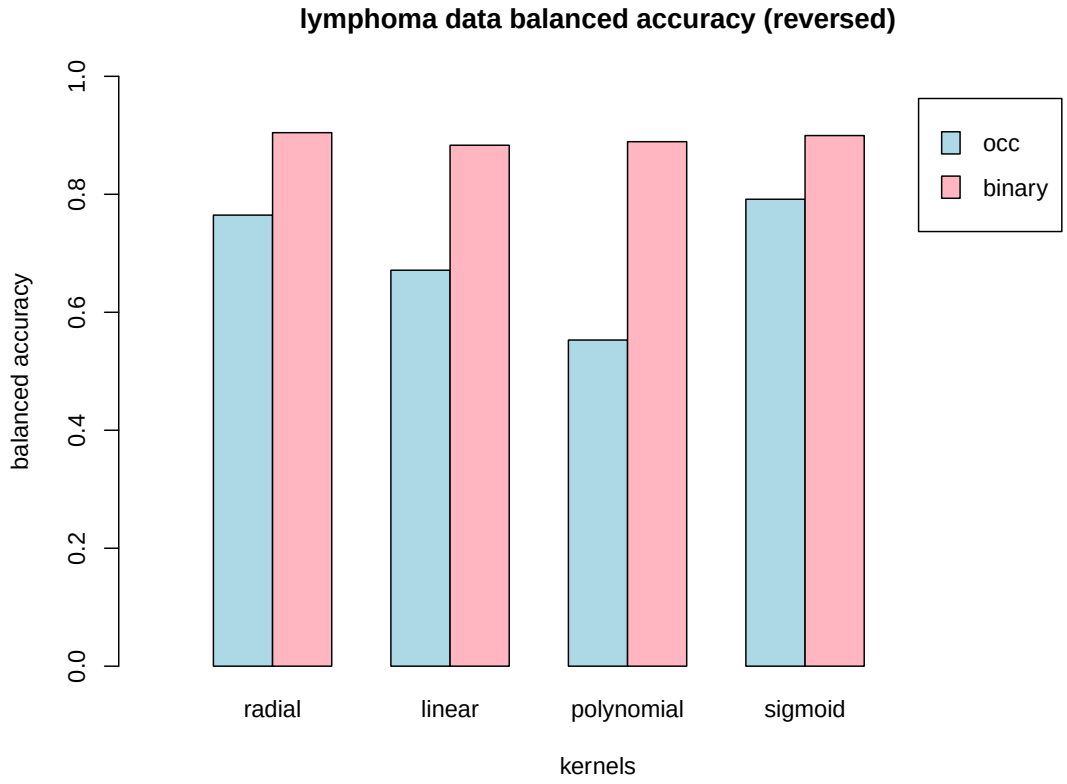

Figure 4.6. *Kernel selection results on the lymphoma data (trained with the test set and test on the training set). The blue bars represent OCC performances, and the pink bars represent for the binary classifier. The performances are evaluated by balanced accuracy. The best BAR of OCC is around 79% (Sigmoid kernel), and the best BAR of binary SVM is 90%.*

them. In order to increase the reliability of our results, we repeat the experiment for ten times for each reduction percentage. Then we draw the output performances as box plots to compare with OCC. Figure 4.7 shows an example workflow of removing 25% negative samples from the training set.

Figure 4.8 and Figure 4.9 show the results of above imbalanceed tests. In these figures, the performances of OCC models are plotted as dotted lines. As we stated above, they are fixed since the OCC training sets always have 0% negatives samples, so we have nothing to remove. The box plots show the performances of binary SVM after removing negative samples. On breast cancer data, the original performance without any removal is slightly better than OCC. But after removing 25% negative samples, OCC performs better. On lymphoma data, the performance of OCC exceeds binary SVM when 75% negative training samples are removed.

We should notice that the original class ratios in these two cancers are different. In breast cancer training set, it is 81:19 (positive:negative). When 25% percent negative samples are removed, the ratio changes to 85:15. Meanwhile in lymphoma training set, the original ratio is 56:44. When 75% percent negative samples are removed, the ratio changes to 84:16. It is quite similar to our result on breast cancer data. For both diseases, the performance of OCC exceeds binary SVM when we reduce the percentage of negative samples to around 15% (class ratio around 85:15). The results may tell us that *at least on these data, OCC SVM performs better than binary SVM when the negative samples only account for around 15% of the entire training set (or even less)*.

4.2 Discussion

In 2005, Yu discussed the limitation of OSVM-based algorithms. According to his comments, OSVM needs a larger number of training samples to generate an accurate decision boundary

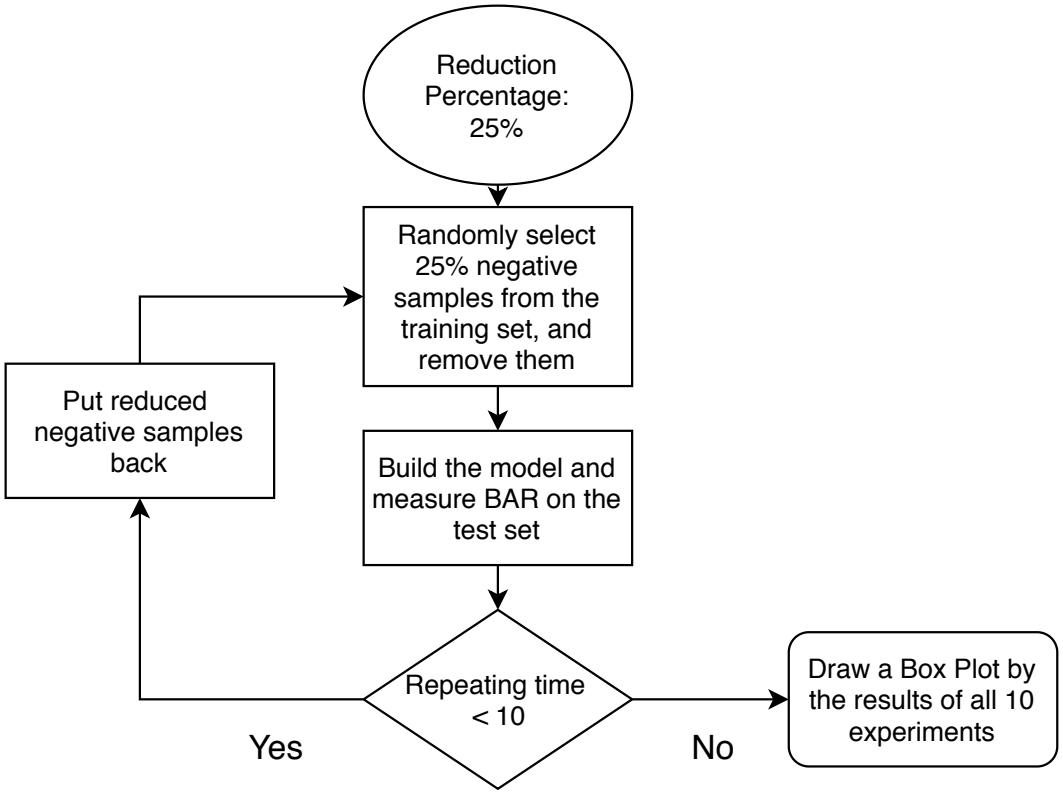

Figure 4.7. *This is an example workflow of removing 25% negative samples in the training set. The first step is to randomly select 25% negative samples and remove them from the training set. Then use the rest samples to train the classifier. Repeat the whole operation for ten times. The removed samples need to be put back before next round. Finally, draw a box plot for all the results.*

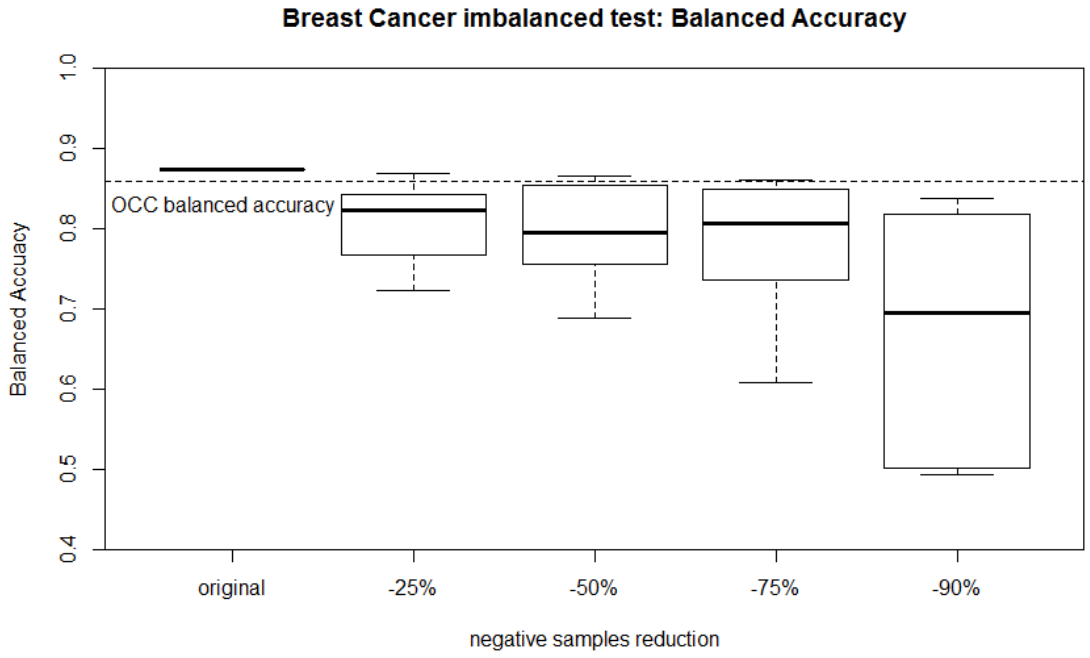

Figure 4.8. *This figures shows the imbalanced test results on breast cancer test sets. The result of OCC model is plotted as dotted line, and the box plots show the results of binary SVM after reducing different percentages of negative samples. It can be seen that OCC exceeded binary SVM after 25% negative samples were removed.*

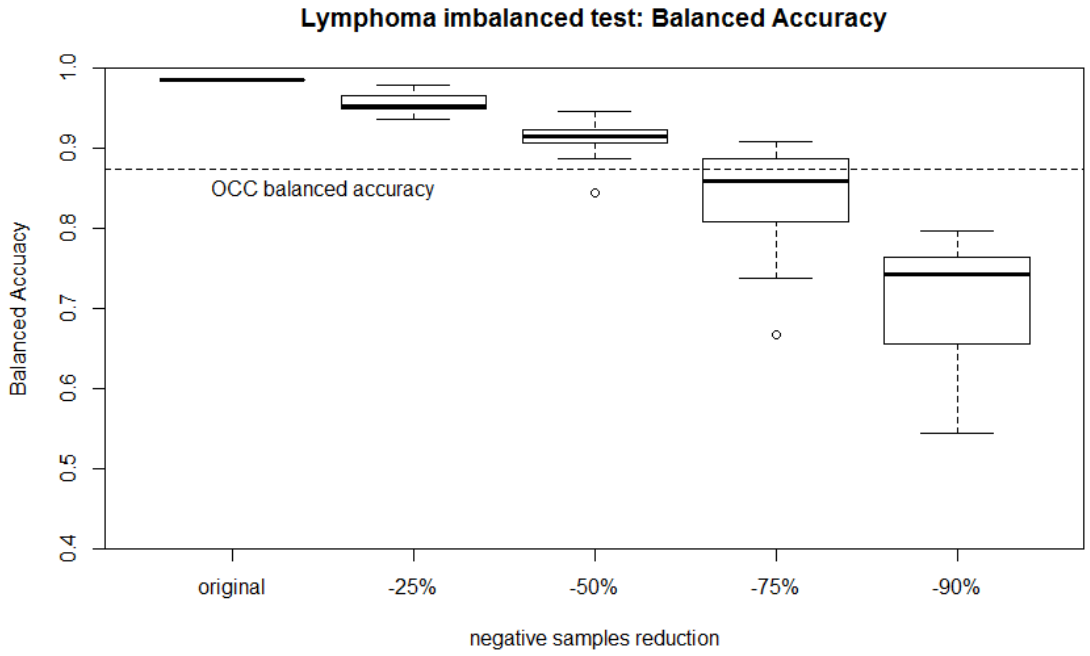

Figure 4.9. *This figures shows the imbalanced test results on lymphoma test sets. The result of OCC model is plotted as dotted line, and the box plots show the results of binary SVM after reducing different percentages of negative samples. It can be seen that OCC exceeded binary SVM after 75% negative samples were removed.*

Table 4.4. *The class ratio (Positive : Negative) of each training set.*

	Original Class Ratio	Class ratio when OCC exceeded Binary SVM
Breast Cancer	81:19	85:15
Lymphoma	56:44	84:16

around the positive sample points [78]. But in practice, recruiting volunteers with a certain disease can be difficult. It is very common that only tens of samples exist in a microarray dataset, especially for those rare diseases. Therefore, it might be difficult to use OSVM on other microarray datasets clinically. But on the other hand, because of the difficulty of finding samples, there could be one class totally missing, or the classes are extremely imbalanced. For example, one may have a large dataset with healthy subjects only, and another small dataset with a few number of patients and healthy persons. In this case, it might be better to use the large dataset to build an OSVM model, instead of using the small dataset to build a binary one. The imbalanced tests in our study also support this hypothesis.

Another difficulty is that a microarray dataset can have more than 10000 features. It is reasonable since microarray measures gene expression and homo sapiens have 20000-30000 genes. But for applying SVM model, whether it is OSVM or binary SVM, tens of thousands dimensions are too many. From the perspective of machine learning, there is a problem called the "curse of dimensionality". It means that the potential feature subspace increases exponentially with the data dimension amount increasing, resulting as an extremely long running time, and the irrelevant features could hide the real outliers [19]. From the perspective of biology, a disease may only relate to a few genes, instead of all of them.

One possible solution proposed by Erfani et al in 2016 is to combine OSVM with Deep Learning [19]. They used deep belief networks (DBNs) to extract robust features from the data, then use these features to train one-class SVM models. However, their solution is focusing on large-scale data while microarray data usually has a small size. Another potential approach is Gene Expression Programming (GEP), a genetic algorithm. Several studies use it as the feature selection method [20][4]. But there are two problems when using it on OCC. Firstly, it is designed for

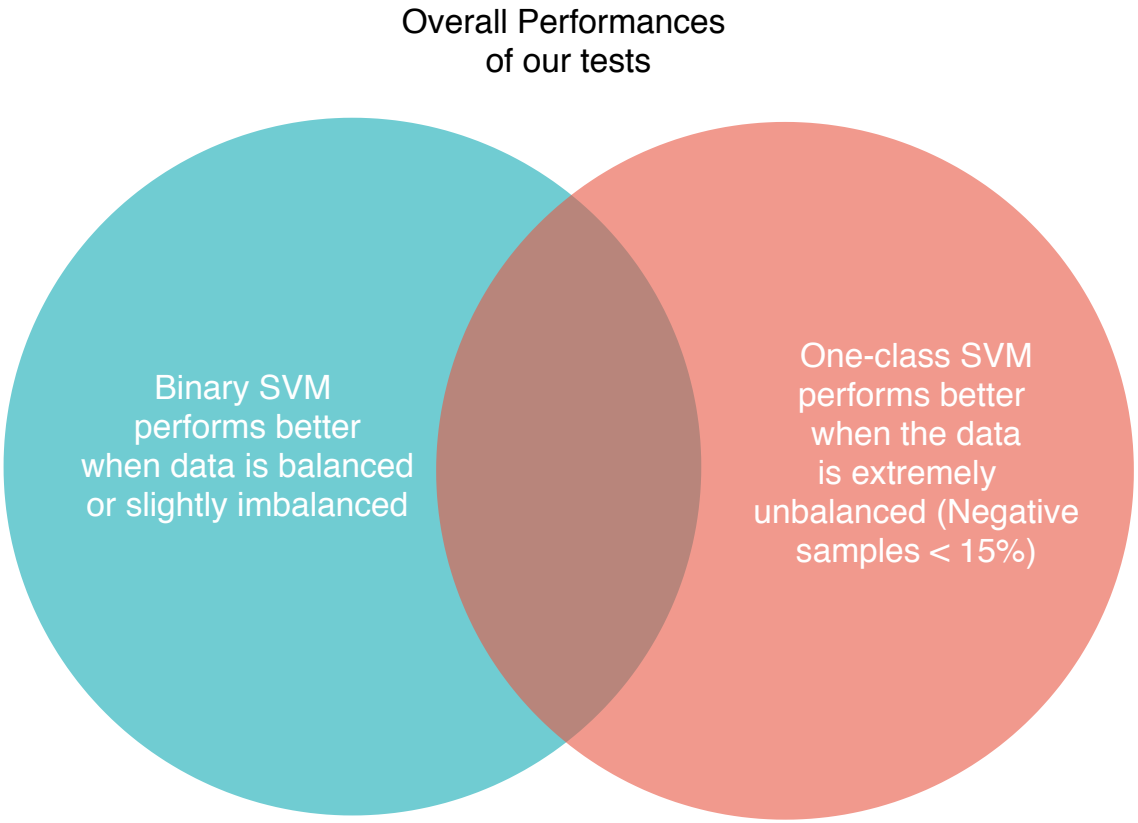

Figure 4.10. *This figure shows the overall performances in our tests. Remind that these are only supported by our experiments of* ***two diseases****, which means it may have different results on other datasets. Based on our results, OCC achieves better results on extremely unbalanced data (microarray data about cancer, negative samples < 15%). For balanced data, e.g. the lymphoma data in our study, binary classifier can have better results. For slightly unbalanced data, they can achieve similar results.*

binary classification methods instead of OCC problem, which may lead to a biased result when the classes are imbalanced. Secondly, GEP can return more than 1000 genes [79]. If we consider about future usage, for instance, the targeted drug research, it is difficult to develop a medicine aiming at 1000 genes.

Therefore, we choose a more conservative approach in our study. Since we are evaluating OCC instead of feature selection methods, we select differentially expressed genes proved by previous studies. It keeps the question clear, and ensure that the results are not affected by feature selection methods. But it is still worth to discuss the feature selection when applying OCC in further study or clinical uses.

We can compare our results with the study of Sokolov et al, who tried to use OCC and binary SVM to distinguish four breast cancer subtypes taxonomy. They evaluated the performances by AUC. Their results showed that the performance of ν-SVM was similar with binary SVM, and they are both better than SVDD. Our study uses balanced accuracy as the evaluation metric and have the experiment on another breast cancer subtype. The results we get are similar to theirs, which support their conclusion from another perspective, that is, whether it is AUC or BAR as the evaluation metric, and no matter what subtype we choose, the ν-SVM has a comparable performance with binary SVM.

In 2016, Zhao et al also built a binary SVM model to classify DLBCL, with the same class taxonomy as us. They used AUC as the evaluation metric, and achieve around 97% for the final performance, which showed that DLBCL data can be classified. To compare with their result, we show that OCC can achieve similar performance by evaluating their balanced accuracies. It gives us the potential to use OCC as a diagnostic tool of DLBCL subtypes.

In our study, we also find that ν-SVM performs better than binary SVM when negative samples only account for around 15% of total (or lower). There is no similar study found. It introduces a

threshold when OCC exceeds the binary. But our results have a limitation, that is, we have only tested two diseases, which are not enough to generate a precise conclusion. More discussion should be done in the future to evaluate the performance of OCC. However, it is no doubt that OCC can achieve good performances on microarray data with proper features and parameters. Figure 4.10 shows the overall performances of our tests.

Now we can go back to our questions at the beginning. We want to figure out if one-class classifiers are useful on gene expression data, how it works when the classes are imbalanced. Furthermore, we want to know if we can apply it on cancer subtype diagnosis. As results, we show that one-class classifier can achieve more than 85% BARs on the data of both diseases. For breast cancer, we use sigmoid kernel and achieve 85.9% BAR; then for lymphoma, we use radial kernel and achieve 88.1% BAR. These performances are quite high. We also show that along with the data becoming more imbalanced, the performance of binary SVM goes down. In our tests, the performance of OSVM exceeds binary SVM when the class ratio is around 85:15 or more imbalanced. All these results can support that one-class SVM has possibility to be a useful machine learning tool to diagnose cancer subtypes, especially when a subtype is extremely rare.

But there are also some conditions that one should check before applying OCC. Firstly, different kernels may have different performances on different datasets. Therefore it is necessary to consider all the kernels at the beginning. Secondly, ν and γ are two very important parameters for OSVM. Therefore, tune the parameters before building the model. Since OCC focuses more on imbalanced data, one should be cautious to select evaluation metric before tuning the parameters. Finally, as we discussed above, the binary SVM is allergic to the imbalance between classes, and OCC may perform better when the data is imbalanced. Therefore, one should check the balance of classes before choosing OCC as a clinical classifier.

5 CONCLUSION

5.1 Study conclusion

In the past chapters, we have evaluated how One-Class SVM works on gene expression data by comparing it with binary SVM. The results show that One-Class classifier can achieve high balanced accuracy on microarray data, especially on imbalanced data. On these cancer datasets, OSVM can achieve better results than binary when the class ratio is around 85:15 or higher. As we mentioned above, traditional classification models need data from more than two classes for training, and one-class methods are usually not as effective as binary models. In our study, we applied OCC in a new field, computational biology, and achieved high performances. This gives more potential to build one-class models in biological, clinical or medical fields. For those patients who have rare diseases or subtypes, OSVM has shown its capacity of classification. Our evaluation supports the possibility that practitioners can recognize rare diseases or uncommon subtypes by gene expression data.

Table 5.1 shows a quick glance of this study. It shows the key ideas, the material we used, major output, and other related information.

5.2 Possible improvement for further studies

Although we used independent datasets as the test sets, we cannot exclude the possibility of coincidence since only two diseases are tested. It should be discussed more in the future study. Another place for improvement is that we select the features by previous studies. In all these studies, they find the differential expressed genes by statistical tests, for example, the student t-test. But for OCC problems, one class can be totally missing, so any traditional statistical tests cannot be applied. For the perspective of machine learning, it would be better to do the feature selection by feature learning algorithms. For instance, SVM recursive feature elimination (SVM RFE) is a potential way to do feature selection before building SVM models [27].

Table 5.1. *A quick glance of the entire study.*

Topic	Evaluating One-Class classifier on gene expression data
Adjusted variable	Gene expression microarray data
Measured variable	Cancer subtypes
Data	Breast cancer and diffuse large B lymphoma microarray data
Algorithm	ν-SVM and binary SVM
Software	R, RStudio
Strengths	High performance, independent test set
Weaknesses	Only two diseases tested
Time spent	6 month

5.3 Challenges

As we mentioned in Chapter 1 and Chapter 2, clinical data can have very small sample sizes. Therefore, building clinical machine learning models can be difficult. Some popular algorithms, e.g. deep learning, may not fit the data size. This is the main challenge in this study. For further biological data analysing, practitioners should pay more attention on selecting algorithms. Usually, Support Vector Machine can be a suitable choice for small size data. Other linear models can also be optional, especially when there are special requirements. For instance, Logistic Regression can be useful when predicting probabilities. Another challenge in this study is the high-dimension problem we mentioned in Chapter 4. A microarray data may have more than 10000 features. Combining the previous "small sample size" problem, the microarray data has the following characteristics: high dimension and small sample size. This makes some algorithms difficult to use. Therefore, it is important to consider about these two problems before analysing.

REFERENCES

[1] C. Abate-Shen. Deregulated homeobox gene expression in cancer: cause or consequence? In: *Nature Reviews Cancer* 2.10 (Oct. 2002), 777–785. URL: http://www.nature.com/articles/nrc907.

[2] B. Alberts, D. Bray, K. Hopkin, A. D. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential cell biology*. 3rd Edition. Garland Science, 2009. Chap. 5, 171–195.

[3] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. Molecular biology of the cell. Garland science. In: *New York* (2007), 1227–1242.

[4] H. Azzawi, J. Hou, R. Alanni, Y. Xiang, R. Abdu-Aljabar, and A. Azzawi. Multiclass Lung Cancer Diagnosis by Gene Expression Programming and Microarray Datasets. In: *ADMA 2017: Advanced Data Mining and Applications*. Springer, Cham, 2017, 541–553. URL: http://link.springer.com/10.1007/978-3-319-69179-4%7B%5C_%7D38.

[5] Bakerstmd. *Wikipedia: Breast cancer*. Wikipedia. 2015. URL: https://en.wikipedia.org/wiki/File:Mammo_breast_cancer_wArrows.jpg.

[6] B. B. Barnes, K. Steindorf, R. Hein, D. Flesch-Janys, and J. Chang-Claude. Population attributable risk of invasive postmenopausal breast cancer and breast cancer subtypes for modifiable and non-modifiable risk factors. In: *Cancer Epidemiology* 35.4 (2011), 345–352. ISSN: 18777821.

[7] C. Bishop. Novelty detection and neural network validation. In: *IEE Proceedings - Vision, Image, and Signal Processing*. Vol. 141. 4. 1994, 217. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=318023.

[8] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006. Chap. 7, 325–358.

[9] M. A. Care, L. Worrillow, R. Patmore, E. Roman, S. Crouch, A. Smith, D. R. Westhead, R. Tooze, A. S. Jack, and S. L. Barrans. Whole genome expression profiling based on paraffin embedded tissue can be used to classify diffuse large B-cell lymphoma and predict clinical outcome. In: *British Journal of Haematology* 159.4 (2012), 441–453.

[10] M. Carlson. *hgu133a.db: Affymetrix Human Genome U133 Set annotation data (chip hgu133a)*. Version R package version 3.2.3. 2016. URL: https://bioconductor.org/packages/release/data/annotation/html/hgu133a.db.html.

[11] B. Carvalho. *pd.hg.u133.plus.2: Platform Design Info for The Manufacturer's Name HG-U133_Plus_2*. Version R package version 3.12.0. 2015. URL: http://bioconductor.org/packages/release/data/annotation/html/pd.hg.u133.plus.2.html.

[12] C.-C. Chang and C.-J. Lin. LIBSVM. In: *ACM Transactions on Intelligent Systems and Technology* 2.3 (2011), 1–27. URL: http://doi.acm.org/10.1145/1961189.1961199.

[13] O. CK, Y. MG, K. W. 3d, and M. WL. The value of estrogen and progesterone receptors in the treatment of breast cancer. In: *Cancer* 46.12 Suppl (1980), 2884–8. URL: https://www.popline.org/node/452041.

[14] E. Clough and T. Barrett. The Gene Expression Omnibus Database. In: *Methods in molecular biology (Clifton, N.J.)* 1418 (2016), 93–110. URL: http://www.ncbi.nlm.nih.gov/pubmed/27008011.

[15] C. Cortes and V. Vapnik. Support-vector networks. In: *Machine Learning* 20.3 (Sept. 1995), 273–297. URL: http://link.springer.com/10.1007/BF00994018.

[16] F. Crick. On protein synthesis. In: *Symp Soc Exp Biol.* 12 (1958), 138–63.

[17] P. Datta. Characteristic concept representations. PhD thesis. 1997.

[18] B. J. Deroo and K. S. Korach. Estrogen receptors and human disease. In: *Journal of Clinical Investigation* 116.3 (2006), 561–570. URL: http://www.jci.org.

[19] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. In: *Pattern Recognition* 58 (Oct. 2016), 121–134. ISSN: 00313203. URL: https://linkinghub.elsevier.com/retrieve/pii/S0031320316300267.

[20] C. Ferreira. Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. In: *Complex Systems* 13.2 (Feb. 2001), 87–129. eprint: 0102027 (cs). URL: http://arxiv.org/abs/cs/0102027.

[21] M. Garcia. Racist in the Machine: The Disturbing Implications of Algorithmic Bias. In: *World Policy Journal* 33.4 (2016), 111–117. URL: https://muse.jhu.edu/article/645268.

[22] V. García, R. A. Mollineda, and J. S. Sánchez. Index of balanced accuracy: A performance measure for skewed class distributions. In: *Iberian Conference on Pattern Recognition and Image Analysis*. Vol. 5524 LNCS. Springer, Berlin, Heidelberg, 2009, 441–448. URL: http://link.springer.com/10.1007/978-3-642-02172-5%7B%5C_%7D57.

[23] A. Gardner, A. Krieger, G. Vachtsevanos, and B. Litt. One-class novelty detection for seizure analysis from intracranial EEG. In: *Journal of Machine Learning Research* 7 (2006), 1025–1044. URL: http://www.jmlr.org/papers/volume7/gardner06a/gardner06a.pdf.

[24] F. M. Giorgi, A. M. Bolger, M. Lohse, and B. Usadel. Algorithm-driven artifacts in median polish summarization of microarray data. In: *BMC bioinformatics* 11 (Nov. 2010), 553. URL: http://www.ncbi.nlm.nih.gov/pubmed/21070630.

[25] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. In: *Science* 286.15 (1999), 531–537. URL: http://science.sciencemag.org/content/sci/286/5439/531.full.pdf.

[26] K. Hempstalk, E. Frank, and I. H. Witten. One-class classification by combining density and class probability estimation. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Vol. 5211 LNAI. PART 1. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 505–519. URL: http://link.springer.com/10.1007/978-3-540-87479-9%7B%5C_%7D51.

[27] M.-L. Huang, Y.-H. Hung, W. M. Lee, R. K. Li, and B.-R. Jiang. SVM-RFE based feature selection and Taguchi parameters optimization for multiclass SVM classifier. In: *TheScientificWorldJournal* 2014 (2014), 795624. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4175386.

[28] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. In: *Biostatistics* 4.2 (2003), 249–264. URL: https://academic.oup.com/biostatistics/article-abstract/4/2/249/245074.

[29] N. Japkowicz. Concept-Learning in the Absence of Counter-Examples : an Autoassociation-Based Approach To Classification. PhD thesis. 1999, 169. URL: https://rucore.libraries.rutgers.edu/rutgers-lib/59019/.

[30] M. K. C. from Jed Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, the R Core Team, M. Benesty, R. Lescarbeau, A. Ziem, L. Scrucca, Y. Tang, C. Candan, and T. Hunt. *caret: Classification and Regression Training*. R package version 6.0-81. 2018. URL: https://CRAN.R-project.org/package=caret.

[31] A. Jemal, A. Robbins, B. Kohler, E. Ward, and C. DeSantis. Childhood and adolescent cancer statistics, 2014. In: *CA: A Cancer Journal for Clinicians* 64.2 (2014), 83–103. ISSN: 00079235.

[32] P. Juszczak, D. M. Tax, E. Pe¸kalska, and R. P. Duin. Minimum spanning tree based one-class classifier. In: *Neurocomputing* 72.7-9 (Mar. 2009), 1859–1869. URL: https://www.sciencedirect.com/science/article/pii/S0925231208003238.

[33] D. E. K Tan, J. Nee Foo, J.-X. Bei, J. Chang, R. Peng, X. Zheng, L. Wei, Y. Huang, W. Yen Lim, J. Li, Q. Cui, S. Hong Chew, R. P. Ebstein, P. Kuperan, S. Thye Lim, M. Tao, S. Hoon Tan, A. Wong, G. Chuan Wong, S. Yong Tan, S. Bian Ng, Y.-X. Zeng, C. Chuen Khor, D. Lin, A. L. H Seow, W.-H. Jia, and J. Liu. Genome-wide association study of B cell non-Hodgkin lymphoma identifies 3q27 as a susceptibility locus in the Chinese population. In: *Nature Genetics* 45.7 (2013), 804–807. ISSN: 1061-4036. URL: https://www.nature.com/articles/ng.2666.pdf.

[34] Y. Karpate, O. Commowick, and C. Barillot. Probabilistic one class learning for automatic detection of multiple sclerosis lesions. In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE, Apr. 2015, 486–489. ISBN: 978-1-4799-2374-8. URL: http://ieeexplore.ieee.org/document/7163917/.

[35] A. K. Kemppinen, J. Kaprio, A. Palotie, and J. Saarela. Systematic review of genome-wide expression studies in multiple sclerosis. In: *BMJ Open* 1.1 (2011), e000053–e000053. ISSN: 2044-6055. URL: http://bmjopen.bmj.com/cgi/doi/10.1136/bmjopen-2011-000053.

[36] S. S. Khan and M. G. Madden. One-class classification: Taxonomy of study and review of techniques. In: *Knowledge Engineering Review* 29.3 (2014), 345–374. URL: https://arxiv.org/pdf/1312.0049.pdf.

[37] F. Letouzey, F. Denis, and R. Gilleron. Learning from positive and unlabeled examples. In: *International Conference on Algorithmic Learning Theory*. Vol. 348. 1. Springer, Berlin, Heidelberg, Dec. 2000, 71–85. URL: http://link.springer.com/10.1007/3-540-40992-0%7B%5C_%7D6.

[38] C. Li and Y. Zhang. Bagging One-Class Decision Trees. In: *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, Oct. 2008, 420–423. URL: http://ieeexplore.ieee.org/document/4666151/.

[39] N. K. LoConte, A. M. Brewster, J. S. Kaur, J. K. Merrill, and A. J. Alberg. Alcohol and Cancer: A Statement of the American Society of Clinical Oncology. In: *Journal of clinical oncology : official journal of the American Society of Clinical Oncology* 36.1 (Jan. 2018), 83–93. ISSN: 1527-7755. URL: http://www.ncbi.nlm.nih.gov/pubmed/29112463.

[40] J. Luo, G. Hu, G. Ni, Z. Pan, and L. Ding. Research on Cost-Sensitive Learning in One-Class Anomaly Detection Algorithms. In: *International Conference on Autonomic and Trusted Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, 259–268. URL: http://link.springer.com/10.1007/978-3-540-73547-2%7B%5C_%7D27.

[41] M. Madden and Y. Liu. One-Class Support Vector Machine Calibration Using Particle Swarm Optimisation. In: *Machine Learning* August (2007), 91–100. URL: http://vmserver14.nuigalway.ie:8080/jspui/handle/10379/204.

[42] M. P. Madigan, R. N. Hoover, J. Benichou, R. G. Ziegler, and C. Byrne. Proportion of Breast Cancer Cases in the United States Explained by Well-Established Risk Factors. In: *JNCI Journal of the National Cancer Institute* 87.22 (Nov. 2002), 1681–1685. URL: https://academic.oup.com/jnci/article-lookup/doi/10.1093/jnci/87.22.1681.

[43] L. M. Manevitz and M. Yousef. Document classification on neural networks using only positive examples (poster session). In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '00*. New York, New York, USA: ACM Press, 2000, 304–306. URL: http://portal.acm.org/citation.cfm?doid=345508.345608.

[44] D. J. McConkey and W. Choi. Molecular Subtypes of Bladder Cancer. In: *Current Oncology Reports* 20.10 (Oct. 2018), 77. ISSN: 1523-3790. URL: http://link.springer.com/10.1007/s11912-018-0727-5.

[45] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-8. 2017. URL: https://CRAN.R-project.org/package=e1071.

[46] T. C. Minter. Single-Class Classification. In: *In Proceedings of Symposium on Machine Processing of Remotely Sensed Data*. 1975, 2A–12–2A–15. URL: http://docs.lib.purdue.edu/lars%7B%5C_%7Dsymphttp://docs.lib.purdue.edu/lars%7B%5C_%7Dsymp/54.

[47] R. D. Morin, M. Mendez-Lago, A. J. Mungall, M. A. Marra, and et al. Frequent mutation of histone modifying genes in non-Hodgkin lymphoma HHS Public Access. In: *Nature* 476.7360 (2012), 298–303. URL: http://www.nature.com/authors/editorial%7B%5C_%7Dpolicies/license.html%7B%5C#%7Dterms%7B%5CS%7D.

[48] F. Mosteller and J. W. Tukey. Data analysis and regression: a second course in statistics. In: *Addison-Wesley Series in Behavioral Science: Quantitative Methods* (1977).

[49] D. T. Munroe and M. G. Madden. Multi-class and single-class classification approaches to vehicle model recognition from images. In: *16th Irish Conference on Artificial Intelligence and Cognitive Science*. 2005, 93–102. URL: https://aran.library.nuigalway.ie/xmlui/handle/10379/191.

[50] D. Murphy. Gene Expression Studies Using Microarrays: Principles, Problems, And Prospects. In: *Advances in Physiology Education* 26 (2002), 256–270. URL: https://www.physiology.org/doi/pdf/10.1152/advan.00043.2002.

[51] Nephron. *Wikipedia: Diffuse large B-cell lymphoma*. Wikipedia. 2010. URL: https://en.wikipedia.org/wiki/File:Diffuse_large_B_cell_lymphoma_-_cytology_low_mag.jpg.

[52] D. Norris FRCPA. WHO Classification of Tumours of Haematopoietic and Lymphoid Tissues (4th Edition) - Lymphoma Classification in the Third Millennium. In: *QML Pathology Newsletter* 4 (2012), 233–243. URL: http://www.qml.com.au/Portals/0/PDF/Newsletters/QML%7B%5C_%7DNL%7B%5C_%7D4%7B%5C_%7D2012.pdf.

[53] K. Pearson. On lines and planes of closest fit to systems of points in space. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1901), 559–572. URL: http://pca.narod.ru/pearson1901.pdf.

[54] E. Pennisi. *DNA study forces rethink of what it means to be a gene*. 2007. DOI: 10.1126/science.316.5831.1556. URL: www.sciencemag.org.

[55] Piotr JUSZCZAK. Learning to recognise. A study on one-class classification and active learning. PhD thesis. 2006, 18–9. URL: http://www.ncbi.nlm.nih.gov/pubmed/21320521.

[56] G. Ritter and M. T. Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. In: *Pattern Recognition Letters* 18.6 (June 1997), 525–539. URL: https://www.sciencedirect.com/science/article/pii/S0167865597000494.

[57] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. In: *IBM Journal of Research and Development* 3.3 (July 1959), 210–229. URL: http://ieeexplore.ieee.org/document/5392560/.

[58] M. Schmidt, D. Böhm, C. Von Törne, E. Steiner, A. Puhl, H. Pilch, H. A. Lehr, J. G. Hengstler, H. Kölbl, and M. Gehrmann. The humoral immune system has a key prognostic impact in node-negative breast cancer. In: *Cancer Research* 68.13 (2008), 5405–5413.

[59] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In: *Advances In Neural Information Processing Systems 12* 12 (2000), 582–588. URL: http://www.cms.livjm.ac.uk/library/archive/Grid%20Computing/NoveltyDetection/sch00support.pdf.

[60] SEER. *Non-Hodgkin Lymphoma - Cancer Stat Facts*. 2011. URL: https://seer.cancer.gov/statfacts/html/nhl.html (visited on 07/23/2019).

[61] A. Senf, X.-w. Chen, and A. Zhang. Comparison of One-Class SVM and Two-Class SVM for Fold Recognition. In: *International Conference on Neural Information Processing*. Springer, Berlin, Heidelberg, 2006, 140–149. URL: http://link.springer.com/10.1007/11893257%7B%5C_%7D16.

[62] J. Silva and R. Willett. Hypergraph-based anomaly detection of high-dimensional co-occurrences. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 31. 3. Mar. 2009, 563–569. URL: http://ieeexplore.ieee.org/document/4626961/.

[63] A. Sokolov, E. O. Paull, and J. M. Stuart. One-Class Detection of Cell States in Tumor Subtypes. In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 21 (2016), 405–416. URL: https://www.worldscientific.com/doi/pdf/10.1142/9789814749411%7B%5C_%7D0037.

[64] B. Stewart and C. Wild. World Cancer Report 2014. In: *International Agency for Research on Cancer* 22 (2014), 362–373, 482–494.

[65] E. Taub, J. M. Deleo, and E. Brad Thompson. Sequential Comparative Hybridizations Analyzed by Computerized Image Processing Can Identify and Quantitate Regulated RNAs. In: *DNA* 2.4 (1983), 309–327. URL: https://www.liebertpub.com/doi/pdf/10.1089/dna.1983.2.309.

[66] D. M. J. Tax and R. P. W. Duin. Support Vector Data Description. In: *Machine Learning* 54 (2004), 45–66. URL: https://link.springer.com/content/pdf/10.1023%7B%5C%%7D2FB%7B%5C%%7D3AMACH.0000008084.60811.49.pdf.

[67] D. M. Tax and R. P. Duin. Support vector domain description. In: *Pattern Recognition Letters* 20.11-13 (Nov. 1999), 1191–1199. URL: https://www.sciencedirect.com/science/article/pii/S0167865599000872.

[68] J. Tian and H. Gu. Anomaly detection combining one-class SVMs and particle swarm optimization algorithms. In: *Nonlinear Dynamics* 61.1-2 (July 2010), 303–310. ISSN: 0924-090X. URL: http://link.springer.com/10.1007/s11071-009-9650-5.

[69] C. A. Tirado, W. Chen, R. García, K. A. Kohlman, and N. Rao. Genomic profiling using array comparative genomic hybridization define distinct subtypes of diffuse large b-cell lymphoma: A review of the literature. In: *Journal of Hematology and Oncology* 5 (2012), 54. URL: http://www.jhoonline.org/content/5/1/54.

[70] S. D. Villalba and P. Cunningham. An evaluation of dimension reduction techniques for one-class classification. In: *Artificial Intelligence Review* 27.4 SPEC. ISS. (2007), 273–294. URL: https://link.springer.com/content/pdf/10.1007%7B%5C%%7D2Fs10462-008-9082-5.pdf.

[71] K. Wang and S. Stolfo. One-Class Training for Masquerade Detection. In: *ICDM Workshop on Data Mining for Computer Security, Melbourne, Florida* (2003), 10–19. URL: https://academiccommons.columbia.edu/doi/10.7916/D89C7455.

[72] X. Wang, Y. Lin, C. Song, E. Sibille, and G. C. Tseng. Detecting disease-associated genes with confounding variable adjustment and the impact on genomic meta-analysis: With application to major depressive disorder. In: *BMC Bioinformatics* 13.1 (Mar. 2012), 52. URL: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-13-52.

[73] J. D. Watson and F. H. C. Crick. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. In: *Nature* 171.4356 (Apr. 1953), 737–738. ISSN: 0028-0836. URL: http://www.nature.com/articles/171737a0.

[74] M. Watson. Assessment of Suspected Cancer. In: *InnovAiT: Education and inspiration for general practice* 1.2 (2008), 94–107. ISSN: 1755-7380. URL: https://journals.sagepub.com/doi/pdf/10.1093/innovait/inn001.

[75] P. Wirapati, C. Sotiriou, S. Kunkel, P. Farmer, S. Pradervand, B. Haibe-Kains, C. Desmedt, M. Ignatiadis, T. Sengstag, F. Schütz, D. R. Goldstein, M. Piccart, and M. Delorenzi. Meta-analysis of gene expression profiles in breast cancer: Toward a unified understanding of breast cancer subtyping and prognosis signatures. In: *Breast Cancer Research* 10.4 (2008). URL: http://breast-cancer-research.com/content/10/4/R65Thisarticleisonlineat:http://breast-cancer-research.com/content/10/4/R65.

[76] M. Yousef, M. D. Saçar Demirci, W. Khalifa, and J. Allmer. Feature Selection Has a Large Impact on One-Class Classification Accuracy for MicroRNAs in Plants. In: *Advances in Bioinformatics* 2016 (2016), 1–6. ISSN: 1687-8027.

[77] H. Yu. SVMC: single-class classification with support vector machines. In: *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*. Vol. 18. Citeseer. 2003, 567–574.

[78] H. Yu. Single-class classification with mapping convergence. In: *Machine Learning* 61.1-3 (Nov. 2005), 49–69. URL: http://link.springer.com/10.1007/s10994-005-1122-7.

[79] J. Zhang, V. Grubor, C. L. Love, S. S. Dave, and et al. Genetic heterogeneity of diffuse large B-cell lymphoma. In: *Proceedings of the National Academy of Sciences of the United States of America*. Vol. 110. 4. National Academy of Sciences, Jan. 2013, 1398–403. DOI: 10.1073/pnas.1205299110. URL: http://www.ncbi.nlm.nih.gov/pubmed/23292937%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3557051.

[80] S. Zhao, X. Dong, W. Shen, Z. Ye, and R. Xiang. Machine learning-based classification of diffuse large B-cell lymphoma patients by eight gene expression profiles. In: *Cancer Medicine* 5.5 (May 2016), 837–852. URL: http://doi.wiley.com/10.1002/cam4.650.

A APPENDICES

```r
parameter_tuning <- function(data, data_start_index,
                             label, gamma, nu, nfolds, folds, type){
  x.in = data
  y.in = label
  acc.out = matrix(0,nrow = length(nu),ncol = length(gamma))
  for(k in 1:length(nu)){
    for(kk in 1:length(gamma)){
      tmp.acc = NULL
      for(i in 1:nfolds){
        testIndexes <- which(folds==i,arr.ind=TRUE)
        testData <- x.in[testIndexes, ]
        testClass <- y.in[testIndexes]
        trainData <- x.in[-testIndexes, ]
        trainClass <- y.in[-testIndexes]
        if(type == "one-classification"){
          x = trainData
          x = x[x$label == 1, ]
          x = x[,data_start_index:ncol(x)]
          tmp.svm = svm(x = x,
                        nu = nu[k],
                        gamma = gamma[kk],
                        type='one-classification',
                        kernel = "radial",
                        scale = TRUE)
          tmp.pred =
            predict(tmp.svm, testData[,data_start_index:ncol(testData)])
          tmp.conf = confusionMatrix(as.factor(as.integer(tmp.pred)),
                                     as.factor(testClass))
        }else if(type == "binary"){
          tmp.svm = svm(x=trainData[data_start_index:ncol(trainData)],
                        y=trainData$label,
                        gamma = gamma[kk],
                        #cost = 10,
                        #kernel = "radial",
                        scale = TRUE)
          tmp.pred =
            predict(tmp.svm, testData[,data_start_index:ncol(testData)])
          confusionMatrix(as.factor(tmp.pred),as.factor(testClass))
          tmp.conf = confusionMatrix(as.factor(tmp.pred),
                                     as.factor(testClass))
        }else{
          print("Wrong svm type input!")
          return(0)
        }
        tmp.acc = c(tmp.acc,tmp.conf$byClass["Balanced Accuracy"])
      }
      acc.out[k,kk] = mean(tmp.acc)
    }
  }
  colnames(acc.out) = as.character(gamma)
  rownames(acc.out) = as.character(nu)
  return(acc.out)
}
```

Program A.1. *the R code of parameter tuning function*